2019 27th European Signal Processing Conference (EUSIPCO)

Deep Log-Likelihood Ratio Quantization

Marius Arvinte, Ahmed H. Tewfik and Sriram Vishwanath
Department of Electrical and Computer Engineering
University of Texas at Austin
Austin, Texas 78712
Email: arvinte@utexas.edu

***Abstract*—In this work, a deep learning-based method for log-likelihood ratio (LLR) lossy compression and quantization is proposed, with emphasis on a single-input single-output uncorrelated fading communication setting. A deep autoencoder network is trained to compress, quantize and reconstruct the bit log-likelihood ratios corresponding to a single transmitted symbol. Specifically, the encoder maps to a latent space with dimension equal to the number of sufficient statistics required to recover the inputs — equal to three in this case — while the decoder aims to reconstruct a noisy version of the latent representation with the purpose of modeling quantization effects in a differentiable way. Simulation results show that, when applied to a standard rate-1/2 low-density parity-check (LDPC) code, a finite precision compression factor of nearly three times is achieved when storing an entire codeword, with an incurred loss of performance lower than 0.15 dB compared to straightforward scalar quantization of the log-likelihood ratios and the method is competitive with state-of-the-art approaches.**

I. Introduction

Quantization is a key component of communication networks and has been intensely researched in the past years. Since practical systems are often memory-limited, quantizing either channel observations or derived information, such as log-likelihood ratios, using as few bits as possible is of critical importance. At the same time, considerable research in deep learning and data science broadly aims to answer the question: *Given high-dimensional data, is there a low-dimensional representation of it*? Considering the log-likelihood ratios derived from transmitting over a memoryless channel, the answer is an immediate *yes*, since, given the channel observation and state information, their values can be derived exactly.

In this paper, we propose a deep learning-based LLR quantization algorithm that empirically answers the question: *Given a set of log-likelihood ratios, is there a low-dimensional representation of it that is robust to quantization*? Our results show that we are able to construct such a highly nonlinear representation by using a deep autoencoder.

Quantization with one-bit analog-to-digital converters has been extensively studied, such as in [1], where the authors present theoretical results regarding the capacity, revealing that the regime of one-bit quantization is benefited by a low operating signal-to-noise ratio (SNR) and increased number of receive antennas. In [2], the effects of quantizing the output of a binary-input discrete memoryless channel are studied and the optimal quantizer for a given number of levels is derived.

Prior work in [3], as well as [4] investigates optimal scalar LLR quantization and designs a quantizer that aims to maximize mutual information, while in [5] the authors design a vector quantizer with the same objective and investigate its performance in a Turbo-coded, hybrid automatic repeat request (HARQ) scenario, demonstrating that an equivalent precision of three bits per value is sufficient to retain most of the performance.

The recent years have seen a resurgence of deep neural networks (DNNs) as universal function approximators [6], as well as a very rapidly increasing interest in using deep learning to solve problems in communication systems [7]. One particularly appealing architecture for communication problems is the *deep autoencoder* [8]. Broadly speaking, an autoencoder aims to approximate the function $f(\mathbf{x}) = \mathbf{x}$ under certain structural constraints placed on its latent representation. Typically, these constraints can be related either to the dimension of the latent space or to the signal structure itself.

Previous work in [9] uses an autoencoder to learn novel, highly nonlinear modulation schemes that offer an implicit coding gain and are robust to channel variations. In another example, the authors of [10] propose an OFDM-autoencoder structure that performs a pair of time/frequency transforms in the latent domain to impose structure on it similar to that of an OFDM signal. Using autoencoders for compression has been extensively studied in computer vision, where they are used for lossy compression of images [11], with results showing that they are competitive with state-of-the-art JPEG compression, as well as being used for image denoising.

Motivated by these recent results, we propose the use of a deep autoencoder for log-likelihood ratio quantization and storage at the receiver side, in a single-input single-output uncorrelated fading channel scenario. We recognize that a latent space of dimension three captures all sufficient statistics and use this in designing the autoencoder. There are three key components to the proposed architecture:

i) A loss function that is weighted towards more accurate reconstruction of low (uncertain) LLR values.
ii) A latent representation of dimension equal to the number of sufficient statistics required to reconstruct the LLR values for each channel use.
iii) A Gaussian noise layer that approximates numerical quantization noise.

During inference, the receiver uses the encoder to compress the vector of LLR values corresponding to a modulated symbol to its latent representation, performs simple uniform scalar quantization and stores the values for future use. The decoder,

978-9-0827-9703-9/19/$31.00 ©2019 IEEE

or a predetermined lookup table, is used to reconstruct the LLR values from this quantized latent representation.

The performance of our proposed method is evaluated when coupled with high-order quadrature amplitude modulation (QAM) and LDPC coding in both standalone and HARQ scenarios, where we show that a compression factor of almost three times is achieved with minimal loss of performance, ultimately enabling more efficient buffer usage for memory-limited applications.

II. Preliminaries

A. System Model

Let s be the transmitted symbol drawn from a constellation $\mathcal{C}$ with cardinality 2^K, corresponding to bits $b_i, i = 1, \ldots, K$. Then, the received symbol r, after passing through a single-input single-output memoryless fading channel, is given by

$$r = hs + n, \tag{1}$$

where $n \sim \mathcal{CN}(0, \sigma_n^2)$ is the complex Gaussian noise and h is the i.i.d. complex channel coefficient. Assuming equal prior probabilities for all bits and that the receiver has complete channel state information, the log-likelihood ratio of bit b_i is given by [12]

$$L_i = \log \frac{P(r|b_i = 1)}{P(r|b_i = 0)} = \log \frac{\sum\limits_{s \in \mathcal{C}, b_i = 1} \exp - \frac{|r - hs|^2}{\sigma_n^2}}{\sum\limits_{s \in \mathcal{C}, b_i = 0} \exp - \frac{|r - hs|^2}{\sigma_n^2}}. \tag{2}$$

Factoring out the h term and letting $\tilde{r} = r/h$ yields

$$L_i = \log \frac{\sum\limits_{s \in \mathcal{C}, b_i = 1} \exp - \frac{|h|^2}{\sigma_n^2} |\tilde{r} - s|^2}{\sum\limits_{s \in \mathcal{C}, b_i = 0} \exp - \frac{|h|^2}{\sigma_n^2} |\tilde{r} - s|^2}. \tag{3}$$

Let

$$G = \left(\frac{|h|}{\sigma_n} \right)^2, \tilde{r}_r = \mathrm{Re}\{\tilde{r}\}, \tilde{r}_i = \mathrm{Im}\{\tilde{r}\}, \tag{4}$$

then the real-valued vector $(G, \tilde{r}_r, \tilde{r}_i) \in \mathbb{R}^3$ is a sufficient statistic to determine the log-likelihood ratios L_i corresponding to a single channel use [12]. Note that G is exactly the instantaneous SNR.

Thus, as long as we have access to, or exactly recover, any bijective function of the vector of real-valued sufficient statistics, the exact log-likelihood ratios can be computed as well. Let

$$\Lambda_i = \tanh \left(\frac{L_i}{2} \right), \tag{5}$$

be the *soft bit* associated to the i-th bit [3], with the immediate property that $\Lambda_i \in [-1, 1]$. This will prove useful in training our deep neural network, since large values are implicitly compressed by this transformation.

B. Deep Autoencoders

Broadly speaking, an autoencoder is a multilayer neural network that aims to approximate the identity function between its inputs and outputs under specific constraints placed on an intermediate output termed *latent representation*. Letting $\mathbf{x}$ be the input to the autoencoder, $\mathbf{y}$ its output and $\mathbf{z}$ the latent representation, the input-output relationship is given by

$$\mathbf{y} = g\big(\mathbf{z}; \theta_g\big) = g\big(f(\mathbf{x}; \theta_f); \theta_g\big), \tag{6}$$

where f and g are the *encoder* and *decoder*, parametrized by weights θ_f and θ_g, respectively. Abusing notation and dropping the weights, the loss function of a general autoencoder, given a dataset $(\mathbf{X})_i, i = 1, \ldots, N$, can be expressed as

$$\mathcal{L}(\mathbf{X}) = \sum_{i=1}^{N} \mathcal{D}\big(\mathbf{x}_i, g(f(\mathbf{x}_i))\big), \tag{7}$$

where $\mathcal{D}$ is a differentiable distance function.

Typical constraints placed on the latent representation include that it is restricted to a low dimensional subspace e.g. $\mathbf{z} \in \mathbb{R}^M$, with $M \ll \dim(\mathbf{x})$, or that it is noised during training, for each sample, to yield

$$\tilde{\mathbf{z}}_i = \mathbf{z}_i + \mathbf{n}_i, \tag{8}$$

where $\mathbf{n}$ is typically i.i.d. Gaussian with adjustable mean and variance [11]. Note that this operation is differentiable and has unit gradient, thus can be used in conjunction with backpropagation during training.

III. The Proposed Scheme

The deep LLR quantization scheme consists of two parts: an autoencoder with a weighted loss function, a latent space of dimension three and a Gaussian noise layer active during training, coupled with a scalar quantization function applied after training is completed. Assuming the soft bits Λ_i are computed for the current symbol using (5), the goal is to store them in finite precision using as few bits as possible, with a performance degradation as small as possible. Since soft bits with low absolute values are more prone to causing error propagation when perturbed in modern decoding algorithms (e.g., [13]), we propose to use as distance function the weighted squared loss

$$\mathcal{D}(\Lambda_i, \tilde{\Lambda}_i) = \sum_{j=1}^{K} \frac{|\Lambda_j - \tilde{\Lambda}_j|^2}{|\Lambda_j| + \epsilon}, \tag{9}$$

where $\epsilon = 10^{-4}$ prevents division by very small amounts and numerical degeneration. Thus, the loss function of the autoencoder becomes

$$\mathcal{L}(\mathbf{\Lambda}, \tilde{\mathbf{\Lambda}}) = \sum_{i=1}^{N} \sum_{j=1}^{K} \frac{|\Lambda_j^{(i)} - \tilde{\Lambda}_j^{(i)}|^2}{|\Lambda_j^{(i)}| + \epsilon}, \tag{10}$$

where $\Lambda_j^{(i)}$ is the j-th soft bit of the i-th training sample and $\tilde{\Lambda}_j^{(i)} = g\big(f(\Lambda^{(i)})\big)_j$ is its autoencoder reconstruction.

This enables the autoencoder to learn more accurate representations of soft bits with low absolute values, and ultimately log-likelihood ratios with the same property, since $\tanh$ is approximately linear around zero. At the same time, emphasis is placed on reconstructing soft bits with absolute value close to 1, where $\mathcal{D}$ becomes closer to the Euclidean distance.

The dimension of the latent representation is constrained to three, since this is the number of sufficient statistics for recovering the log-likelihood ratios, as given by (4). Using an additive Gaussian noise layer with zero mean and variance σ_t^2, the autoencoder is trained to learn a robust latent representation of the soft bits. Incorporating quantization directly in the training procedure is not possible in a straightforward manner because the step function has a null derivative almost everywhere, hence is not compatible with modern gradient descent training methods. In this sense, we effectively approximate the quantization error with Gaussian noise.

The last layer of the encoder and the decoder are designed to use $\tanh$ as activation function. For the decoder, it is straightforward to do so, since it helps constrain the outputs in the $[-1, 1]$ interval, where the input signal is found. For the encoder, the reasoning behind this is twofold:

i) $\tanh$ serves as an implicit regularization counter to adding Gaussian noise. If the output of the encoder was unbounded (e.g. linear or ReLU activation), then the autoencoder would learn a robust $\mathbf{z}$ by simply increasing its energy indefinitely.

ii) Constraining $\mathbf{z}$ to lie in $[-1, 1]^3$ makes it more amenable to subsequent finite precision quantization, allowing the use of the same quantizer for all components z_i.

Once training is complete, the soft bit vector $\mathbf{\Lambda}$ corresponding to a symbol is input to the encoder, resulting in the latent representation $\mathbf{z} = (z_1, z_2, z_3)$. $\mathbf{z}$ is elementwise quantized to finite precision using the function

$$Q(x) = \begin{cases} \operatorname{sgn}\{x\}\delta & \text{if } |x| > \delta \\ \frac{\delta}{2^{N_b}}\Delta_k & \text{if } x \in -\delta + [\frac{k\delta}{2^{N_b-1}}, \frac{(k+1)\delta}{2^{N_b-1}}] \end{cases}, \quad (11)$$

where $\Delta_k = -2^{N_b} + 2k + 1$. Thus, all values outside the range $[-\delta, \delta]$ are first saturated to $\operatorname{sgn}\{x\}\delta$ and subsequently uniformly quantized using N_b bits.

Reconstructing the soft bits from the latent representation $\mathbf{z}$ is done by simply passing it through the decoder g. Thus, the reconstructed soft bits, as a function of the original values, take the form

$$\tilde{\mathbf{\Lambda}} = g\big(Q(f(\mathbf{\Lambda}))\big). \quad (12)$$

Fig. 1 shows the architecture of the proposed scheme, both during training and inference time. We use a symmetrical architecture for the autoencoder, in which the encoder and decoder have the same number of hidden layers, with their output dimensions mirrored.

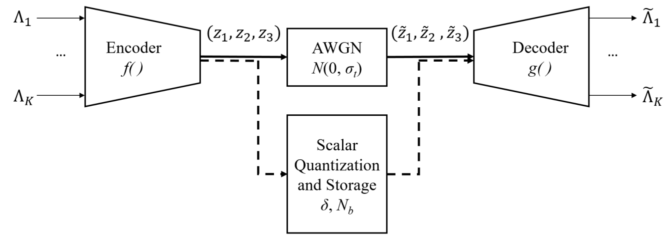

Fig. 1. The architecture of the proposed scheme. Solid lines correspond to the signal path during training, where $\tilde{\mathbf{z}}$ represents the noisy latent representation of the input soft bits. Dashed lines correspond to the inference path, where the same notation represents the quantized latent representation of $\mathbf{z}$, which is stored for future use.

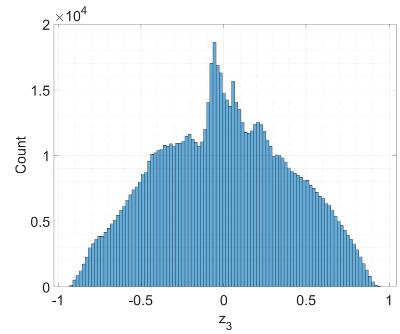

Fig. 2. Empirical distribution of z_3 at an SNR value of 18 dB, 256-QAM. Obtained by encoding the LLR values from $10,000$ LDPC codewords.

IV. Experimental Results

A. Architecture and Training Details

We use a number of three feedforward, fully-connected hidden layers for the encoder and decoder each, respectively, with an intermediate output size of $4K$ and a latent representation of size three. All the hidden layers use the rectified linear unit (ReLU) function as activation except for the final layers of the encoder and decoder, where we use $tanh$. We use the Adam [14] optimization algorithm to train the autoencoder, with a learning rate of 10^{-3} and a batch size of $65\,536$ samples. For training and validation data, we generate random bit payloads, encode them using LDPC, interleave, modulate and apply an i.i.d. Rayleigh fading channel (i.e., $h \sim \mathcal{CN}(0, 1)$) for each QAM symbol. We compute the exact expressions of the log-likelihood ratios by using (2) and use them as training data. We do this for multiple SNR values, generating $10\,000$ codewords for each one, and concatenate the resulted values together, allowing the proposed architecture to work without requiring separate SNR estimates as input. We train the architecture with the loss function in (10) for a number of 2000 epochs.

Once the autoencoder is trained, we design our latent space uniform quantizer. Fig. 2 plots the marginal distribution of the latent signal z_3 when LLR values at a specific SNR are encoded. Empirically, we choose $\delta = 0.8$ as clipping threshold

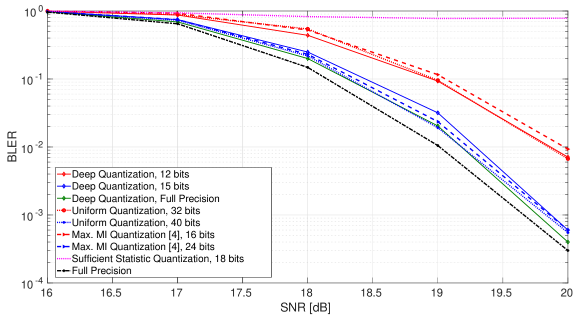

Fig. 3. Block error rate performance for the single transmissions scenario, using 256-QAM modulation. For each SNR value, 10 000 LDPC codewords are simulated, separate from the training data used by the deep quantization method.

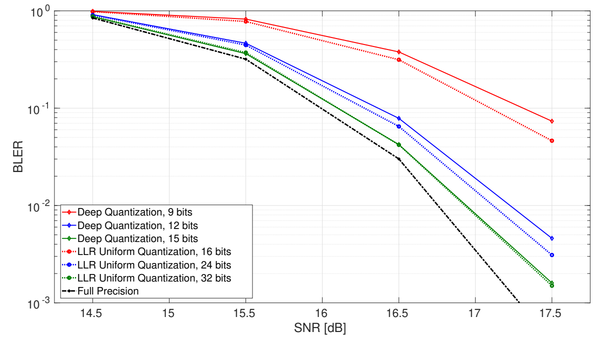

Fig. 4. Block error rate performance for the multiple, overlapping transmissions scenario, using 256-QAM modulation. For each SNR value, 10 000 LDPC codewords are simulated, separate from the training data used by the deep quantization method.

for all latent signals. The simulations were performed using the Keras library [15], as well as Matlab. The source code is available online[1], alongside extended simulation results.

B. Single Transmission Performance

The block error rate performance is simulated in a single transmission scenario. Since prior work on LLR quantization shows that a number of 4-5 bits usually retains most of the decoding performance [3], our architecture is capable of achieving a higher compression ratio as we target more densely packed constellations. For the reference quantization schemes we consider two alternatives:

1) A simple saturation and uniform quantization algorithm with low computational complexity. Empirically, we saturate LLR values outside the $[-4, 4]$ interval since it provides the best block error rate performance.
2) The maximum mutual information quantizer from [4]. This method is a data-driven approach able to derive the optimal quantization levels for each individual LLR in a high-order QAM scheme. We also use initial codebook values in the $[-3, 3]$ interval to further boost the performance of this method.

We also simulate the performance of applying scalar quantization to the vector $(G, \tilde{r}_r, \tilde{r}_i)$ to highlight the very large error floor it creates when limiting the number of bits. In particular, we know that G has a Rayleigh distribution for our channel model, thus we apply the optimal scalar quantizer in [16] for it. Fig. 3 shows the performance results obtained. Comparing at 10^{-2} block error rate, we notice there is a performance loss of 0.15 dB incurred by the autoencoder in full precision. Since submitting this work, we have been successful in covering this performance gap through different considerations. A preprint of our work is available at [17].

Using the data in Fig. 2, the mean and variance of z_3 are estimated to be approximately 0 and 0.158, respectively. Coupled with the injected noise power of $\sigma_t^2 = 10^{-6}$, the effective SNR in the latent space is approximately 52 dB. This appears large compared to the actual noise introduced by quantizing to 3-5 bits, since it corresponds to an effective resolution of 8 bits, but we find that, if σ_t is further increased, the autoencoder does not converge to a solution with low reconstruction error, since too much noise is added to the latent representation. Fig. 3 shows that quantization of the latent representation and subsequent reconstruction retains the performance of the reference schemes achieving a compression factor of up to 2.7 times compared to the simple scalar quantization method and being competitive with the state-of-the-art methods.

C. Multiple Transmission Performance

The same autoencoder and channel code are used in a HARQ scenario with two transmissions, each sending a random, but fixed, half of the codeword and an additional one third of bits from the other half, having an effective code rate of $3/8$. The log-likelihood ratios of the first transmission are quantized, while the ones from the second transmission are assumed to be available in full precision. The log-likelihood ratios common to both transmissions are combined using equal gain combining, given by the expression

$$L_i = \tilde{L}_i^{(1)} + L_i^{(2)}, \tag{13}$$

where $\tilde{L}_i^{(1)}$ is the reconstructed version of the quantized ratio $L_i^{(1)}$ and $L_i^{(2)}$ is the full precision LLR from the second transmission.

Fig. 4 shows the performance results obtained with a 256-QAM modulation, where we notice that the proposed 15 bit representation has near identical performance to 32 bits in the reference scalar implementation, achieving a compression ratio of 2.13 times, at a performance very close to full precision. The comparison with the state-of-the-art method is similar to that in Fig. 3, thus we omit it from this plot. Alternatively, if a more significant loss of performance is tolerated, deep quantization can be used with 9 bits and a compression ratio of 1.77 times, approaching the hard-output decoding limit of one equivalent bit per LLR value.

[1] https://github.com/mariusarvinte/deep-llr-quantization

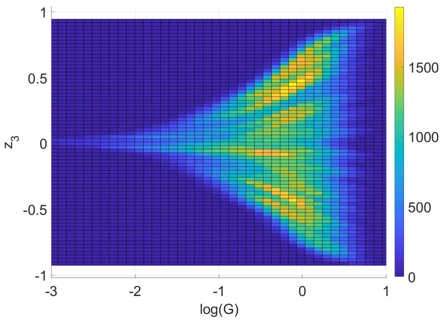

Fig. 5. Empirical joint distribution of $(\log G, z_3)$ in the same conditions as Fig. 2. The intensity of the color is proportional to the probability mass of the respective bin.

D. Discussion

We now provide an interpretation of what the autoencoder learns for its representation. Since the latent space dimension is constrained to the number of sufficient statistics that describe the log-likelihood ratios, it is reasonable to assume that any valid (i.e., with reconstruction error sufficiently small) representation is approximately a bijective, potentially highly nonlinear, function of the statistics

$$(z_1, z_2, z_3) \approx \Phi(G, \tilde{r}_r, \tilde{r}_i). \tag{14}$$

Training with added Gaussian noise in the latent space drives the autoencoder to learn a function Φ of the sufficient statistics that is robust to quantization noise, assuming the approximation is reasonable. Note that the identity mapping is not a suitable function in this case, since quantizing $G = |h|^2/\sigma_n^2$, in particular, introduces a severe error floor, as seen in Fig. 3.

Thus, the learned mapping Φ is potentially a highly nonlinear function. This is indeed the case when visualizing the empirical joint distribution between each sufficient statistic and each latent signal component. Fig. 5 plots the joint distribution of $\log(G)$ and z_3, where it is observed that this is not a one-to-one mapping, thus z_3 contains information on at least one more sufficient statistic. While this provides an interpretation of what the autoencoder learns for a Rayleigh fading channel, the structure is fragile to changing the distribution of h, where retraining the entire structure is required. This can be overcome by applying techniques from the field of machine learning under covariate shift, which is left for future work.

Finally, note that since the latent space is quantized in a finite number of points, this is also true for the reconstructed log-likelihood ratios. Thus, if desired, one can precompute all possible reconstructions, store them in a larger, external memory and eliminate the decoder from the inference stage.

V. Conclusions

In this paper, a deep learning-based log-likelihood ratio quantization algorithm for uncorrelated fading channels was introduced. It is recognized that the number of sufficient statistics required to recover the LLR values is equal to three and an autoencoder is designed and trained to extract a latent representation of that exact dimension, while being robust to quantization noise when recovering the LLR values.

Using an autoencoder and a uniform quantization function, we have shown that we can achieve compression factors of 2.7 and two versus simple scalar quantization for standalone and HARQ scenarios, respectively, when a standard rate-1/2 LDPC code is used in conjuction with 256-QAM modulation. At the same time, our method is competitive with state-of-the-art quantization algorithms. Other advantages are a width of the hidden layers on the order $\mathcal{O}(K)$ and the fact that the same autoencoder is trained and used across a wide range of SNR values. The proposed scheme has a wide range of applications, including, but not limited to, HARQ scenarios, relay and feedback channels.

References

[1] J. Mo and R. W. Heath, Jr., “Capacity Analysis of One-Bit Quantized MIMO Systems With Transmitter Channel State Information,” *IEEE Trans. Signal Processing*, vol. 63, no. 20, pp. 5498–5512, Oct. 2015.

[2] B. M. Kurkoski and H. Yagi, “Quantization of Binary-Input Discrete Memoryless Channels,” *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 4544–4552, Aug. 2014.

[3] W. Rave, “Quantization of Log-Likelihood Ratios to Maximize Mutual Information,” *IEEE Signal Processing Letters*, vol. 16, pp. 283–286, 2009.

[4] A. Winkelbauer and G. Matz, “On quantization of log-likelihood ratios for maximum mutual information,” in *2015 IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2015, pp. 316–320.

[5] M. Danieli, S. Forchhammer, and J. D. Andersen, “Maximum Mutual Information Vector Quantization of Log-Likelihood Ratios for Memory Efficient HARQ Implementations,” in *Data Compression Conference*, 2010.

[6] K. Hornik, M. Stinchcombe, and H. White, “Multilayer Feedforward Networks are Universal Approximators,” *Neural Networks*, vol. 2, pp. 359–366, 1989.

[7] C. Zhang, P. Patras, and H. Haddadi. (2018) Deep Learning in Mobile and Wireless Networking: A Survey. [Online]. Available: https://arxiv.org/abs/1803.04311

[8] P. Baldi, “Autoencoders, Unsupervised Learning, and Deep Architectures,” in *ICML Unsupervised and Transfer Learning*, 2012.

[9] T. J. O’Shea, K. Karra, and T. C. Clancy, “Learning to Communicate: Channel Auto-encoders, Domain Specific Regularizers, and Attention,” in *Proc. IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 2016.

[10] A. Felix, S. Cammerer, S. Dorner, J. Hoydis, and S. ten Brink, “OFDM-Autoencoder for End-to-End Learning of Communications Systems,” in *Proc. International Workshop on Signal Processing Advances in Wireless Communications (SPAWC’18)*, Kalamata, Greece, Jun. 2018.

[11] L. Theis, W. Shi, A. Cunningham, and F. Huszar, “Lossy Image Compression with Compressive Autoencoders,” in *ICLR*, 2017.

[12] R. G. Gallager, *Stochastic Processes: Theory for Applications*. Cambridge University Press, 2014.

[13] X. Zhang and P. H. Siegel, “Quantized Min-Sum Decoders with Low Error Floor for LDPC Codes,” in *Proc. International Symposium on Information Theory 2012 (ISIT’12)*, Jul. 2012.

[14] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[15] (2019) Keras: The Python Deep Learning library. [Online]. Available: https://keras.io/

[16] W. Pearlman and G. Senge, “Optimal Quantization of the Rayleigh Probability Distribution,” *IEEE Trans. Communications*, vol. 27, no. 1, pp. 101–112, Jan. 1979.

[17] M. Arvinte, S. Vishwanath, and A. H. Tewfik, “Deep learning-based quantization of l-values for gray-coded modulation,” *Preprint*, 2019. [Online]. Available: http://arxiv.org/abs/1906.07849