Accelerating Wireless Channel Autoencoders for Short Coherence-time Communications

Manuel Eugenio Morocho-Cayamcela and Wansu Lim

Abstract: **Traditional wireless communication theory is based on complex probabilistic models and fixed conjectures, which limit the optimal utilization of spectrum resources. Deep learning has been used to design end-to-end communication systems using an encoder to replace the transmitter and a decoder for the receiver. We address the challenge to update the parameters of a wireless channel autoencoder (AE) under a time-varying channel with short coherence-time. We suggest an optimized training algorithm that updates the learning rate value on a per-dimension basis, restricting the past gradients instead of accumulating them. We also scale the initial weights of our AE by sampling them from a normalized uniform distribution. While recently proposed AE configurations might fail to converge at a few number of epochs, our setting attains a fast convergence maintaining its robustness to large gradients, oscillations, and vanishing problems. By simulation results, we demonstrate that our proposed AE configuration improves the bit reconstruction accuracy in shorter training time.**

Index Terms: **Autoencoders, channel estimation, deep learning, physical layer, wireless systems.**

I. INTRODUCTION

THE design and implementation of conventional communication systems are built upon strong probabilistic models and assumptions [1]. Furthermore, they are limited in explaining the theory to practice when handling the complexity of optimization for new wireless applications with high degrees of freedom. Deep learning (DL) has shown a high potential to address these challenges via data-driven solutions, improving the utilization of limited wireless spectrum resources [2]–[4]. Instead of following a rigid design, new generations of wireless systems empowered by cognitive radio can learn from data, and optimize their spectrum utilization to enhance their performance. These smart communication systems rely on various detection, classification, and prediction tasks, such as signal detection and signal type identification for spectrum sensing. To address these tasks, DL provides powerful automated means for communication systems to learn from spectrum data and adapt to its dynamics [5]–[7]. Wireless communications data come in large volumes at high rates, and are subject to interference and security threats due to the shared nature of the medium [8], [9]. Traditional modeling often fall short when capturing the delicate relationship between highly complex spectrum data, whereas DL has a robust capacity to meet the requirements (i.e., data rate, mobility, latency, connection density, energy efficiency, traffic capacity, etc.) of the next-generation mobile and wireless communication systems (see [10]–[13] and references therein).

DL-based designs for the wireless communications physical layer has been studied recently, and one of the most promising among them is the channel autoencoder (AE), that interprets a traditional end-to-end wireless communications system as a pair of convolutional neural networks (CNNs). A convolutional AE can be described as a deep neural network (DNN) architecture that consists of an encoder that learns a latent representation of the given data, and a decoder that reconstructs the input data from the encoded data. In this setting, joint modulation and coding at the transmitter correspond to the encoder, and joint decoding and demodulation at the receiver correspond to the decoder. Initial *time-invariant* simulations have demonstrated the promising capacity of AEs by optimizing the reconstruction of information bits through artificial neural network's impairment layers. However, the assumption that the wireless channel is time-invariant, is only feasible when the receiver motion is less than $\lambda/2$, where λ is the wavelength represented as the ratio of the speed of light c to the carrier frequency f_c. Emerging wireless applications are far from the latter assumption, requiring a precise *time-varying* channel estimation, where the transmitter, receiver, and objects in the propagation environment may move relative to one another with a non-linear relation. If we let the receiver velocity to be expressed as v (measured in meters/second), the *coherence-time* can be expressed as $T_c = \lambda/2v$ (measured in seconds) [9]. High-mobility scenarios reduce the T_c of the wireless system (i.e., the time during which the wireless channel can be assumed as time-invariant), demanding a more frequent update of the AE's parameters without sacrificing accuracy and convergence time.

A reduced T_c limits the numbers of training epochs the AE has to reach convergence (i.e., to be able to fully capture the dense representations of the wireless channel impairments). Limiting the number of epochs of the AE, compels to increase the learning rate of the optimizer to cope with the same signal-to-noise ratio (SNR) vs. block error rate (BLER). The problem with increasing the learning rate is that it can make the AE fail to converge, diverge, or stop learning after a few iterations. If the AE fails to converge, the end-to-end wireless system will produce errors in the reconstruction of the bits. We can verify that the AE has not converged by analyzing the epoch vs. ac-

Manuscript received December 20, 2019; revised March 5, 2020; approved for publication by Chan-Byoung Chae, Guest Editor, May 24, 2020.

This work was supported by the National Research Foundation of Korea (NRF) under grant 2017R1C1B5016837; and in part by the Technology Development Program (S2829065) funded by the Ministry of SMEs and Startups (MSS, Korea).

M. E. Morocho-Cayamcela is with the Department of Electronic Engineering, Kumoh National Institute of Technology, Gumi, Gyeongsangbuk-do, 39177 South Korea, e-mail: eugeniomorocho@kumoh.ac.kr.

W. Lim is with the Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi, Gyeongsangbuk-do, 39177 South Korea, e-mail: wansu.lim@kumoh.ac.kr.

W. Lim is the corresponding author.

Digital Object Identifier: 10.1109/JCN.2020.000011

1229-2370/19/$10.00 © 2020 KICS

Table 1. Summary of wireless channel autoencoder related works.

Methodology	Data input	Channel	Cost function	Optimizer	Performance metric	Future work	Reference
Introduced the concept of a channel AE to communicate binary information over an impaired channel.	Discrete bits (i.e., 0's and 1's).	Random Gaussian variable to each in-phase (I), and quadrature (Q) samples, as well as frequency and phase offset.	Cross-entropy loss.	RMSprop and Adam, but obtained better results with Adam.	SNR vs. BER.	Rely on channel gradient approximation methods, and constrained by error feedback bandwidth and latency.	T. O'Shea *et al.* (2016) [14].
Design a DL-based physical layer for multiple input multiple output (MIMO) wireless communications using an AE.	A one-hot input vector of length 2^k with a single non-zero value of 1 ($k =$ bits).	Additive white Gaussian noise $N(0, \sigma)$.	Categorical cross-entropy loss function.	Adaptive momentum.	SNR vs. BER.	Further comparison with error correction coding baselines may be added.	T. O'Shea *et al.* (2017) [15].
Interpret a communications system as an end-to-end reconstruction task with an AE to learn full transceivers implementation for a given channel model.	Input symbol represented as a *one-hot* vector.	Additive noise layer with a fixed variance $\beta = (2RE_b/N_0)^{-1}$.	Categorical cross-entropy between input symbol and the recovered message with the highest probability $\ell_{CE}(\boldsymbol{\theta})$.	Stochastic gradient descent (SGD).	BLER, i.e., Pr($\hat{s} \neq s$) of the communications system.	Has only been validated by simulations for block-based transmissions. Limited scalability to long block lengths.	T. O'Shea *et al.* (2017) [16].
Extend the existing ideas toward continuous end-to-end data transmission over-the-air, which eases the restriction to short block lengths on [16].	An *embedding* function that takes an integer input i and returns the ith column of a matrix.	Additive white gaussian noise (AWGN) with fixed random noise power σ^2 per complex symbol.	Cross-entropy loss function $L_{loss} = -log(b_s)$.	Stochastic gradient descent (SGD).	BLER.	Does not enable on-the-fly finetuning to adapt to varying channel conditions for which it has not been trained	S. Dörner *et al.* (2018) [17].
End-to-end communication system using AEs to capture channel impairments, jointly optimizing the transmitter and receiver operations in a single-antenna, multiple-antenna, and multi-user communications scenarios.	Input symbol represented as a *one-hot* vector.	Additive noise layer with a fixed variance $\beta = (2RE_b/N_0)^{-1}$.	Categorical cross-entropy $\ell_{CE}(\boldsymbol{\theta})$.	Stochastic gradient descent (SGD). Adam (lr = 0.001).	BLER to be reduced at the receiver.	The channel models used to simulate the impairments are Gaussian.	T. Erpek *et al.* (2019) [18].

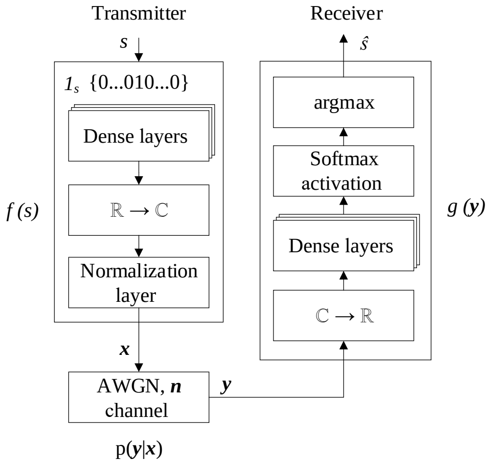


Fig. 1. An autoencoder-based end-to-end communication system.

curacy, and epoch vs. loss responses. Also, a slow convergence rate will be translated into a higher BLER, and spacing in the constellations produced by the AE.

In this paper, we address the need to update the parameters of the wireless channel AE under a *time-varying* channel with short coherence-time. Our goal is to structure a wireless channel AE that can converge within a fewer number of epochs to cope with the need of a rapid-varying real-world environment. We suggest an optimized learning-rate update algorithm that changes its value on a per-dimension basis, restricting the past gradients instead of accumulating them. Furthermore, we scale the initial parameters of the AE by sampling them from a normalized uniform distribution. The proposed convolutional encoder-decoder design is able to capture wireless channel impairments by jointly optimizing the transmitter and receiver operations. We demonstrate that whereas previously proposed AE configurations might fail to converge at a few numbers of epochs, our setting attains fast convergence, retaining robustness to large gradients, oscillations, and vanishing gradients. By simulations, we compare the SNR vs. BLER, and learned constellations from different wireless channel AE configurations. We show that our proposed AE reaches a higher bit reconstruction accuracy in shorter training time. Our results demonstrate the power of optimization strategies and proper weight initialization in providing means to meet the requirements of new high-mobility and multiple-environmental wireless applications.

II. RELATED WORKS

The concept of learning an end-to-end communications system by using an *encoder* to replace the transmitter functionalities such as modulation and coding, and a *decoder* for the receiver functionalities such as demodulation and decoding was first introduced by T. O'Shea *et al.* in [14]. The authors used a feed-forward network (FFN) to replace the functions of the transmitter, and a decoder to act as the receiver. The channel noise was simulated by adding a random Gaussian variable to each in-phase (I) and quadrature (Q) samples, as well as frequency and phase offset. They used discrete bits (i.e., 1's and 0's) as the input of the AE, and compared the SNR against the bit error rate (BER). Subsequently in [15], the authors generalized the AE from [14] to multiple input multiple outputs (MIMO) wireless communications using a *one-hot* vector as input, noise from an additive white Gaussian distribution $N(0, \sigma)$. The encoder and decoder were built using a convolutional FFN. They used the same performance metric as the previous work. In [16], the authors used an additive noise layer operating at rate $R = 4/7$ with a fixed variance $\beta = (2RE_b/N_0)^{-1}$ (with E_b representing the energy per bit and N_0 the noise power spectral density), to supersede the wireless channel effects, with a similar AE architecture as [14], [15]. The input block was compared with the recovered message via a categorical cross-entropy function. They show that the BLER obtained with their AE can approximate the results of a conventional wireless communications system. S. Dörner *et al.* [17], extended the existing ideas to continuous end-to-end data transmission over-the-air which eases the restriction to short block lengths found on [16]. They used an *embedding* function that takes an integer as input i and returns the ith column of a matrix. Additive white Gaussian noise (AWGN) with fixed random noise power σ^2 was added to each complex symbol. The authors used the standardized cross-entropy loss function $L_{\text{loss}} = -log(b_s)$ to compare the sent blocks b_s to the reconstructed ones. T. Erpek *et al.* [18], generalized the end-to-end AE system to jointly optimize the transmitter and receiver operations in a single-antenna, multiple-antenna, and multi-user communications scenarios. The authors used the *one-hot* vector coding to represent the input information. Analogous to [16], they used an additive noise layer with a fixed variance $\beta = (2RE_b/N_0)^{-1}$ between the encoder and decoder, and a categorical cross-entropy. They showed that the BLER is reduced at the receiver for all the scenarios.

All these works treat a communication system as an unsupervised learning problem (i.e., no additional labels exist in the dataset since the same input is used to compute the loss function), for which stochastic gradient descent (SGD) [19] is a prevalent selection to find the local/global minimum of the categorical cross-entropy loss function. However, SGD does not always guarantee good convergence for this kind of cost function. Moreover, there is no information on the value of the learning rate, which makes it difficult to reproduce the exact results. The learning rate is one of the most sensitive hyperparameters, since it significantly affects the performance of the AE. Further optimizers has been proposed to solve certain challenges of training neural network models, such as root mean square propagation (RMSprop) [20], adaptive gradient (Adagrad) [21], adaptive moment (Adam) [22], Adam-based infinity norm (Adamax) [22], and Nesterov Adam (Nadam) [23]. Nevertheless, either they require to manually select hyperparameters such as the learning rate, or make the learning rate infinitesimally small and stop the parameter update. Lastly, none of the authors have discussed the initialization of the parameters from the FFN. Initializing the weights of the network randomly, or with a value of zero has been proved to cause vanishing and exploding gradient problems [24].

III. AN END-TO-END COMMUNICATION SEQUENCE WITH DEEP LEARNING

The established communication system includes a transmitter, a receiver, and a channel that carries the information between the transmitter and the receiver. *Claude E. Shannon* in its original paper on communication theory [25], stated that the fundamental problem of communication systems is: *"Reproducing at one point either exactly or approximately a message selected at another point"*. That statement is equivalent to the concept of a modern AE, where its job is to reconstruct a given input at its output. In this section, we revisit the physical layer of a conventional communication system design and reformulate it as an end-to-end reconstruction task that aims to optimize the transmitter and receiver components in a single operation.

A. The Limitation of Conventional Communication Systems

Conventional communication systems are divided into multiple independent blocks for the transmitter and receiver. These independent pieces are optimized individually for different tasks [26]. Each block at the transmitter prepares the signal to the effects of the communication channel and noise at the receiver. The source encoder compresses the input data and removes redundancy. The channel encoder adds redundancy to the output of the source encoder in a controlled way. The modulator changes the characteristics of the signal based on the required data rate. The transmitted signal is then distorted and attenuated by the channel. On top of that, the impairments of the receiver's hardware introduce extra noise to the signal. The transmitter processes are reversed at the receiver to recover the information. The optimization of these individual processing blocks is known to be suboptimal, given that it does not optimize the overall system collectively [17]. In this conventional communication system, the transmitter communicates one from the M available messages $s \in \mathcal{M} = \{1, 2, \cdots, M\}$ to the receptor, making n uses of the channel. The transmitter applies the modulation $f : \mathcal{M} \mapsto \mathbb{R}^n$ to the message s, and generates the signal $\boldsymbol{x} = f(s) \in \mathbb{R}^n$ to be transmitted. Digital modulation maps the input symbols from a discrete alphabet to complex numbers that represent the points on the constellation diagram. The process of digital modulation in conventional communication systems has fixed and pre-established constellation diagrams. The desired data rate determines the constellation scheme and the grouping of the input bits for symbol construction. Linear decision regions make it simple to decode the information at the receiver.

B. Accelerating the Convergence of an End-to-End Wireless Communication Autoencoder

Algorithm 1 Parameter learning and optimization

```
Input: Layer inputs $m$, layer outputs $n$, decay rate $\rho$,
    constant $\epsilon$.
Output: Optimal hyperparameters $\boldsymbol{\theta}_{t+1}$.
    Initialisation :
    Initialize $\boldsymbol{\theta} \sim U\left(-\sqrt{\frac{6}{m+n}}, \sqrt{\frac{6}{m+n}}\right)$.        ▷ Glorot.
    Initialize accum. variables $E[g^2]_0 = 0$, $E[\Delta\boldsymbol{\theta}^2]_0 = 0$.
    Define the categorical cross-entropy loss function.
    $\ell_{CE}(\boldsymbol{\theta}) = -\frac{1}{M}\sum_{i=1}^{M}\sum_{j=0}^{2^{kN_t}-1} p'_{o,j}\ log(p_{o,j})$
    Calculate parameter update.
    while stopping criterion not met do        ▷ Num. of epochs.
        for $t = 1 : T$ do
            Sample a minibatch $\mathbb{B}$ of $M$ samples
            Compute gradient: $g_t$
            Accumulate gradient:
                $E[g^2]_t = \rho E[g^2]_{t-1} + (1-\rho)g_t^2$
            Compute update:
                $\Delta\boldsymbol{\theta}_t = -\frac{\text{RMS}[\Delta\boldsymbol{\theta}]_{t-1}}{\text{RMS}[g]_t} g_t$
            Accumulate updates:
                $E[\Delta\boldsymbol{\theta}^2]_t = \rho E[\Delta\boldsymbol{\theta}^2]_{t-1} + (1-\rho)\Delta\boldsymbol{\theta}_t^2$
            Apply update: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \Delta\boldsymbol{\theta}_t$
        end for
    end while
    return $\boldsymbol{\theta}_{t+1}$
```

As opposed to the independent block optimization of conventional communication systems, DL is capable to jointly optimize multiple communications blocks at the transmitter and receiver by training them as DNNs. In an AE system, the output constellation diagrams are not pre-defined but *learned*, based on the desired performance metric to be minimized at the receiver (i.e., the symbol error rate, coherence-time, distance, propagation loss, etc.). The hardware of the transmitter imposes the following constraints (1) [18]:

$$\begin{cases} \|\boldsymbol{x}\|_2^2 \le n & \text{for} \quad \text{energy,} \\ |x_i| \le 1\ \forall i & \text{for} \quad \text{amplitude,} \\ \mathbb{E}\left[|x_i|^2\right] \le 1\ \forall i & \text{for} \quad \text{average power,} \end{cases} \tag{1}$$

on $\boldsymbol{x}$. The data rate of this system is computed as $R = k/n$ (bit/channel use), where $k = log_2(M)$ represents the number of input bits. The notation (n, k) implies that a communication system sends one from the $M = 2^k$ messages (i.e., k bits) over n channel uses. Figure 1 illustrates a block diagram of the channel AE, where the learning process exploits the distribution of the communication channel data under impairments. The communication channel is explained by the density of the conditional probability $p(\boldsymbol{y}|\boldsymbol{x})$, where $\boldsymbol{y} \in \mathbb{R}^n$ denotes the signal at the receiver. The transmitted message s is detected as $\boldsymbol{y}$ at the receiver, where the operation $g : \mathbb{R}^n \mapsto \mathcal{M}$ is applied to estimate the value of $\hat{s}$. The channel AE parameters are optimized to map $\boldsymbol{x}$ to $\boldsymbol{y}$ to enable s to be recovered by minimizing the probability of error. The input symbol s is encoded as a *one-hot* vector, that is, s can only take legal combinations of values with a single high '1' bit and all the others low '0' to allow a state machine to run at a faster clock rate than any other encoding. Determining the state of a one-hot vector has a low and constant cost of accessing one flip-flop. The transmitter is composed of an FNN with two dense layers. The first dense layer has the same number of neurons as the available messages M, and is activated by a rectified linear unit (ReLU). The second dense layer has the same number of neurons as complex baseband symbols n, without activation function to allow negative values. The last dense layer output is reshaped to represent two complex numbers with real (in-phase, I) and imaginary (quadrature, Q) parts for each

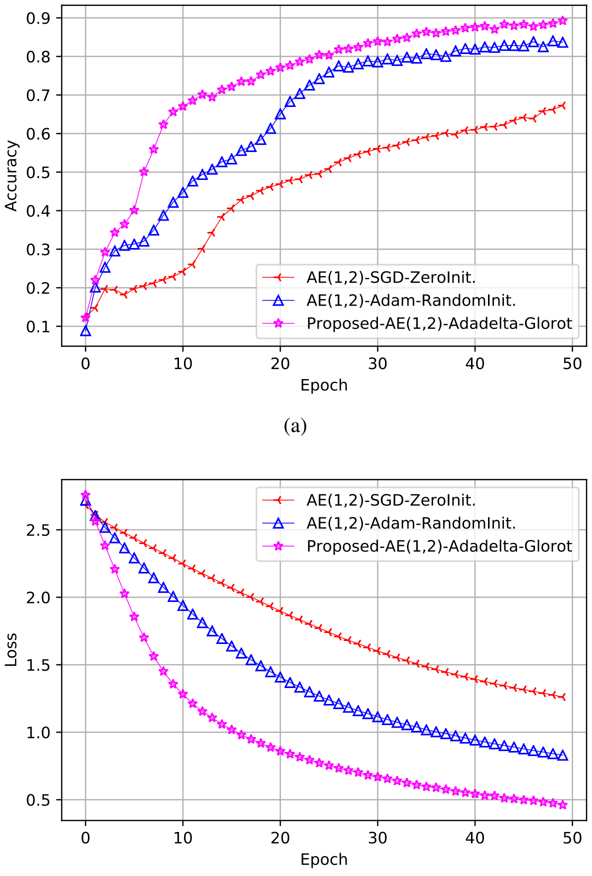


Fig. 2. Training results of our proposed configuration: (a) Accuracy and (b) loss obtained after our autoencoder has been trained for 50 epochs.

modulated input symbol. The final normalization layer of the transmitter ensures that physical power constraints (1) on $\boldsymbol{x}$ are met, preventing the AE to learn unnecessary large outputs and become unstable. The channel is represented by additive white Gaussian noise (AWGN) with variance

$$\beta = (2RE_b/N_0)^{-1}, \tag{2}$$

where E_b/N_0 constitutes the energy per bit E_b to noise power spectral density N_0 ratio. The noise varies for every training example, and it is used for the forward pass to distort the transmitted signal, but neglected in the backward pass. Similar to the transmitter, the receiver consists of a complex to real value transformation, followed by a dense FNN with two layers. The first dense layer has n neurons with ReLU activation, and the second dense layer has M neurons without activation function. The dense layers are followed by a *softmax* activation that outputs the probability vector $\mathbf{p} \in (0,1)^M$ over all possible messages. Lastly, the *argmax* function selects the element of $\mathbf{p}$ with the highest probability value as $\hat{s}$. During training, we use a *categorical cross-entropy loss function* $\ell_{CE}(\boldsymbol{\theta})$ between the transmitter and receiver (3), that jointly optimize them and determine the weights and biases for both of the FNNs that minimize the reconstruction loss.

$$\ell_{CE}(\boldsymbol{\theta}) = -\frac{1}{M}\sum_{i=1}^{M}\sum_{j=0}^{2^{kN_t}-1} s\ log(\hat{s}) \tag{3}$$

The optimization involves the task of minimizing the function $\ell_{CE}(\boldsymbol{\theta})$, by updating $\boldsymbol{\theta}$ in an iterative manner. We denote $\ell'_{CE}(\boldsymbol{\theta})$ as the derivative of our cost function (3). The derivative $\ell'_{CE}(\boldsymbol{\theta})$ gives the slope of $\ell_{CE}(\boldsymbol{\theta})$ at the point $\boldsymbol{\theta}$ (i.e., it defines how to scale the input to obtain the equivalent change in the output $\ell_{CE}(\boldsymbol{\theta}+\epsilon) \approx \ell_{CE}(\boldsymbol{\theta}) + \epsilon\,\ell'_{CE}(\boldsymbol{\theta})$). For a small enough ϵ, the following condition is met:

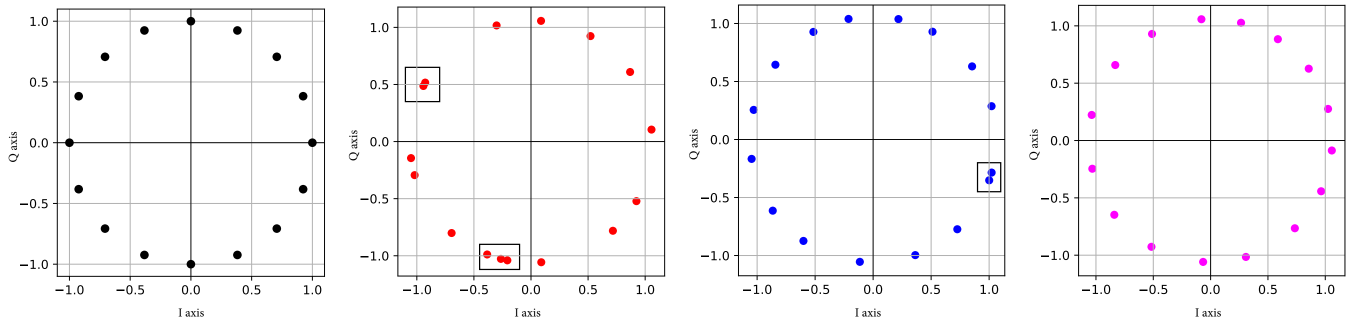
$$\ell_{CE}\left(\boldsymbol{\theta} - \epsilon\,\mathrm{sgn}\left(\ell'_{CE}(\boldsymbol{\theta})\right)\right) < \ell_{CE}(\boldsymbol{\theta}). \tag{4}$$

We aim to minimize our $\ell_{CE}(\boldsymbol{\theta})$ function with multiple inputs. Nevertheless, the theory of minimization allows one scalar output. For our multiple-input function, we use the concept of *partial derivatives* and generalize to the case where the derivative is with respect to a vector, denoting $\nabla_{\boldsymbol{\theta}}\ell_{CE}(\boldsymbol{\theta})$ as the *gradient* containing the partial derivatives of $\ell_{CE}(\boldsymbol{\theta})$. The loss function $\ell_{CE}(\boldsymbol{\theta})$ is minimized by moving $\boldsymbol{\theta}$ in small steps with the opposite sign of the derivative [19]. SGD updates the parameters after computing the gradient of the error with respect to a single training example. This is the optimization method used for almost all the related works in Section II. SGD updates the value of the parameter $\boldsymbol{\theta}$ until convergence (i.e., when every element of the gradient is zero, or very close to zero) as follows:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \epsilon\nabla_{\boldsymbol{\theta}}\ell_{CE}(\boldsymbol{\theta}), \tag{5}$$

where ϵ is the learning rate, a positive scalar that determines the size of the step. This value of ϵ is not evident for the AE design and is considered an essential hyperparameter to reproducible research. Accelerating the convergence of the AE requires using a higher value of ϵ, however by using a high learning rate the algorithm might overshoot the global minimum, or diverge [27] (as will be shown in Section IV). It has been proved that DL architectures based on FFNs, such as an AE, are strongly affected by the choice of ϵ [23]. The value of ϵ can determine whether the end-to-end AE converges or not. Provided that the AE converges with a given value of ϵ, the chosen value can determine the speed of convergence and if it converges to a high or low-cost point. To accelerate the convergence time of an AE and cope with the emerging high-mobility and short coherence-time applications in wireless communications, the optimization strategy is essential. To damp oscillations in directions of high curvature, algorithms that compute the exponentially weighted average of the past gradients and use the new gradient to update the weights have been proposed [28]. Nonetheless, the latter algorithm introduces an additional hyperparameter that needs to be chosen manually. Based on [29], and to avoid the gradients become infinitesimally small, we restricted the past gradients to a *window* to become a local estimate using recent gradients. This method ensures progress in learning even after several iterations. To implement the window, we accumulate an exponentially decaying average of the squared gradients. At time t, we compute this running average $E[g^2]_t$ as follows:

$$E[g^2]_t = \rho E[g^2]_{t-1} + (1-\rho)g_t^2, \tag{6}$$



(a) (b) (c) (d)

Fig. 3. 16PSK constellations generated by: (a) A conventional wireless communication system (16PSK-Hamming(7,4) HD), (b) AE optimized with SGD with zero initialization (16PSK-AE-SGD-Zero [16]–[18]), (c) AE optimized with Adam and initialized randomly (16PSK-AE-Adam-Rand. [14], [15]), and (d) our convolutional AE with the adaptive learning rate and normalized uniform distribution initialization (Proposed AE-Adadelta-Glorot).

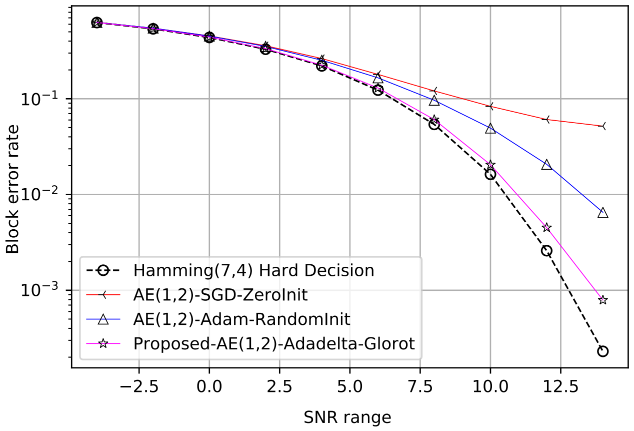


Fig. 4. Signal to noise ratio (SNR) vs. block error rate (BLER) for different convolutional autoencoder configurations after 50 epochs.

where ρ is the decay constant, and g_t is the gradient of the parameters at the tth iteration $\frac{\partial \ell_{CE}(\boldsymbol{\theta}_t)}{\partial \boldsymbol{\theta}_t}$. The root mean square (RMS) of the previous gradients up to time t can be computed with:

$$\text{RMS}[g]_t = \sqrt{E[g^2]_t}. \tag{7}$$

And the resulting parameter update $\Delta\boldsymbol{\theta}_t$ becomes:

$$\Delta\boldsymbol{\theta}_t = -\frac{\text{RMS}[\Delta\boldsymbol{\theta}]_{t-1}}{\text{RMS}[g]_t} g_t. \tag{8}$$

A summary of the wireless channel AE configuration for training can be found in Algorithm 1. The information on how the weights of the FFNs were initialized is absent from previous works, hence we replicated the baseline AE with both, random and zero initialization. Since our AE uses a softmax activation function, we scale the initial weights of our AE fully connected layers with m inputs and n outputs as:

$$\theta_{i,j} \sim U\left(-\sqrt{\frac{6}{m+n}}, \sqrt{\frac{6}{m+n}}\right), \tag{9}$$

by sampling each weight from a uniform distribution $U(-\frac{1}{\sqrt{m}}, \frac{1}{\sqrt{m}})$. This normalized initialization derived from the variance of the uniform distribution avoids the vanishing and exploding gradient problem [24]. The training batch is the set of all possible messages $s \in \mathcal{M}$, and the gradient is derived from a categorical cross-entropy loss function between s and $\hat{s}$.

IV. SIMULATION RESULTS AND PERFORMANCE EVALUATION

Like any other unsupervised learning method, an AE learns without any prior knowledge. The joint optimization is how we force the AE to extract only the features that are necessary and characterize the input data to store it in the bottleneck layer (i.e., the layer that contains the smaller and dense representations). According to [15], an AE can achieve equivalent performance as the Hamming (7, 4) code with maximum likelihood decoding (MLD). The AE achieves the same BLER as uncoded BPSK for a (2, 2) system, and outperforms uncoded BPSK for an (8, 8) system. We have reproduced the latter results and let the AE learn a heavily tailored compression scheme for the specific communication system. The AE configurations are trained using 4 NVIDIA GTX 1080Ti with local parallel pooling. All results are obtained after 50 epochs. Fig. 2(a) reveals that our proposal attains a higher accuracy across the 50 training epochs. Fig. 2(b) shows the rapid loss decrease of our wireless channel AE configuration. Table 2 displays the training and testing time of the models under comparison using an Intel i7-7700 CPU, 4 NVIDIA GTX 1080Ti GPUs in parallel, and the online Cloud TPU v3 from Google. Note that the proposed AE model achieves the fastest training time when tested under the three processor units. Also, our model requires a lower number of epochs to reach convergence. In particular, note that when using a TPU, our model training time is compliant with short coherence-time channel use-cases for 5G scenarios [10]. As an example, for a carrier frequency $f_c = 2$ GHz, and a receiver velocity $v = 108$ km/h (located outdoors), the coherence-time can be as short as $T_c = 2.5$ ms, making it even shorter for higher frequencies. The AEs have learned 16PSK with a random rotation, without any prior modulation knowledge (Fig. 3). 16PSK

Table 2. Training time obtained with different AE configurations and computational processing units.

AE configuration	CPU[†] (s/epoch)	GPU[‡] (s/epoch)	TPU[§] (s/epoch)	Convergence in epoch
SGD-Zero [16]–[18]	5.84×10^{-3}	4.36×10^{-3}	3.45×10^{-3}	160
Adam-Rand [14], [15]	4.43×10^{-3}	2.98×10^{-3}	2.02×10^{-3}	90
Proposed AE[*]	$\mathbf{3.12\times10^{-3}}$	$\mathbf{1.47\times10^{-3}}$	$\mathbf{0.01\times10^{-3}}$	**50**

[†] Trained using Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz. [‡] Trained using 4 NVIDIA GTX 1080Ti with local parallel pooling.

[§] Trained using Cloud TPU v3 in Google Colab. [*] Convolutional AE with dynamic learning rate and normalized initial weights distribution.

constellations generated by (a) a conventional wireless communication system, (b) AE optimized with SGD with zero initialization, (c) AE optimized with Adam and random initialization, and (d) the proposed convolutional wireless channel AE with the adaptive learning, rate and normalized uniform distribution initialization. It is noteworthy that the separation between the constellation points in Figs. 3(b) and 3(c) is not uniformly equal, which increases the BLER at the receiver. The plot of SNR vs. BLER of the wireless channel AE configurations under study can be seen in Fig. 4. We note that SGD optimizer with zero initialization, and Adam with random initialization fail to recover the information blocks from the transmitter. Our adaptive learning with restricted gradients and normalized initialization improves the performance for short coherence-time scenarios.

V. CONCLUSIONS AND FUTURE WORK

We have review how DL architectures can help in the optimization of wireless communication systems. First, the formulation of a transmitter and receiver as an AE for the physical layer has been discussed. An end-to-end optimization is employed for the reconstruction loss, instead of optimizing the individual blocks of a conventional communication system (i.e., synchronization, symbol estimation, error correction, channel coding, modulation, etc.). It has been demonstrated that this formulation enables to capture channel impairments of single-antenna systems, and can match modulation baselines by applying DNNs. We have tackled the challenge of parameters update on a wireless channel AE under a time-varying channel with short coherence-time, by using a dynamic learning rate that updates its value on a per-dimension basis. Also, we have scaled the initial weights of the wireless channel AE by sampling the parameters from a normalized uniform distribution to avoid fading gradients. By simulation results, we show that our proposed wireless channel AE configuration effectively increases the bit reconstruction accuracy in shorter training time. Future works in the field include channel generalization by scaling from an AWGN model to sophisticated real-world channels. This channel generalization might be studied by combining generative with discriminative RF models, in an adversarial way to improve them together. Additionally, researchers may leverage the propagation and physics theory to propose better impairment models. Moreover, additional AEs may be employed to extend this approach to multi-user systems and multiple-antenna systems. It would be interesting to see the new solutions for using AEs as we scale systems. Finally, this work may be transferred to specific domains, like satellite communications, backhaul radios, dense urban wireless, B5G MIMO, etc. All things considered, there is still a wide opportunity for future researchers to include engineering knowledge to exploit AEs in an effort to take wireless communications optimization to a fully-driven DL system.

REFERENCES

[1] M. E. Morocho-Cayamcela, J. N. Njoku, J. Park, and W. Lim, “Learning to communicate with autoencoders: Rethinking wireless systems with deep learning,” in *Proc. IEEE ICAIIC*, 2020, pp. 308–311.

[2] M. E. Morocho-Cayamcela and W. Lim, “Finding the optimal path for V2V multi-hop connectivity with Q-learning and convolutional neural networks,” in *Proc. KICS*, June 2019, pp. 294–297.

[3] L. Wang and D. T. Delaney, “QoE oriented cognitive network based on machine learning and SDN,” in *Proc. ICCSN*, June 2019, pp. 678–681.

[4] M. E. Morocho-Cayamcela and W. Lim, “Proposed cost function using wireless propagation for self-organizing networks,” in *Proc. KICS*, Nov. 2019, pp. 172–174.

[5] M. E. Morocho-Cayamcela, H. Lee, and W. Lim, “Machine learning to improve multi-hop searching and extended wireless reachability in V2X,” *IEEE Commun. Lett.*, early access, 2020.

[6] M. E. Morocho-Cayamcela, M. Maier, and W. Lim, “Breaking wireless propagation environmental uncertainty with deep learning,” *IEEE Trans. Wireless Commun.*, early access, 2020.

[7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, 1st ed., T. Dietterich, Ed. London, England: The MIT Press, 2016.

[8] A. Osseiran, J. F. Monserrat, and P. Marsch, *5G mobile and wireless communications technology*, 1st ed. United Kingdom: Cambridge University Press, 2017.

[9] T. L. Marzetta, E. G. Larsson, H. Yang, and H. Q. Ngo, *Fundamentals of Massive MIMO*, 1st ed. Cambridge, United Kingdom: Cambridge University Press, 2016.

[10] M. E. Morocho-Cayamcela, H. Lee, and W. Lim, “Machine learning for 5G/B5G Mobile and wireless communications: Potential, limitations, and future directions,” *IEEE Access*, vol. 7, pp. 137 184–137 206, Sept. 2019.

[11] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. C. Chen, and L. Hanzo, “Machine learning paradigms for next-generation wireless networks,” *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98–105, Apr. 2017.

[12] F.-L. Luo, *Machine Learning for Future Wireless Communications*, 1st ed. West Sussex,UK: Wiley & Sons, Inc., 2020.

[13] M. E. Morocho-Cayamcela and W. Lim, “Artificial intelligence in 5G Technology: A survey,” in *Proc. ICTC*, 2018, pp. 860–865.

[14] T. J. O’Shea, K. Karra, and T. C. Clancy, “Learning to communicate: Channel auto-encoders, domain specific regularizers, and attention,” in *Proc. IEEE ISSPIT*, Dec. 2016, pp. 223–228.

[15] T. J. O’Shea, T. Erpek, and T. C. Clancy, “Physical layer deep learning of encodings for the MIMO fading channel,” in *Proc. Allerton*, Oct. 2017, pp. 76–80.

[16] T. O’Shea and J. Hoydis, “An Introduction to deep learning for the physical layer,” *IEEE Trans. Cognitive Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[17] S. Dorner, S. Cammerer, J. Hoydis, and S. t. Brink, “Deep learning based communication over the air,” *IEEE J. Selected Topics Signal Process.*, vol. 12, no. 1, pp. 132–143, Feb. 2018.

[18] T. Erpek, T. J. O’Shea, Y. E. Sagduyu, Y. Shi, and T. C. Clancy, “Deep learning for wireless communications,” in *Development and Analysis of Deep Learning Architectures*. Springer, 2020, pp. 223–266.

[19] M. A. Cauchy, “Méthode générale pour la résolution des systèmes d’équations simultanées,” *Comptes Rendus Hedb. Séances Acad. Sci.*, vol. 25, no. 10, pp. 536–538, 1976.

[20] A. Graves, “Generating Sequences With Recurrent Neural Networks,” *arXiv:1308.0850v5*, Aug. 2013.

[21] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” *J. Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[22] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” in *Proc. ICLR*, Dec. 2015.

[23] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, “On the importance of initialization and momentum in deep learning,” in *Proc. ICML*, Feb. 2013, pp. 2176–2184.

[24] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” *Proc. AISTATS*, vol. 9, pp. 249–256, 2010.

[25] C. E. Shannon, “A mathematical theory of communication,” *Bell System Technical J.*, vol. 27, no. 3, pp. 379–423, Oct. 1948.

[26] E. Björnson, J. Hoydis, and L. Sanguinetti, *Massive MIMO Networks*, 1st ed. Pisa, Italy: now Publishers Inc., 2019, vol. 1.

[27] A. Ng, *Machine Learning Yearning: Technical Strategy for AI Engineers in the Era of Deep Learning*. deeplearning.ai, 2018.

[28] N. Qian, “On the momentum term in gradient descent learning algorithms,” *Neural Netw.*, vol. 12, no. 1, pp. 145–151, 1 1999.

[29] M. D. Zeiler, “ADADELTA: An adaptive learning rate method,” *arXiv:1212.5701v1*, Dec. 2012.

Manuel Eugenio Morocho-Cayamcela received the B.S. degree in Electronic Engineering from Universidad Politécnica Salesiana, Cuenca, Ecuador, in 2012 and the M.Sc. degree in communications engineering and networks from The University of Birmingham, England, United Kingdom, in 2016. He is currently working towards the Ph.D. degree in electronic engineering at Kumoh National Institute of Technology, Gumi, South Korea.

From 2017, he has been a Research Assistant with KIT Future Communications and Systems Laboratory. His research interests include communications engineering and networks, artificial intelligence, signal processing, statistical analysis, and optimization.

Mr. Morocho-Cayamcela was a recipient of the SENESCYT Fellowship from The National Secretariat for Higher Education, Science, Technology and Innovation of Ecuador in 2015, the KIT Doctoral Grant from Kumoh National Institute of Technology in 2017, and the Best Paper Award at ICNGC 2017, and KICS 2019. Mr. Morocho-Cayamcela is a member of the Institute of Electrical and Electronics Engineers (IEEE), and the Korean Institute of Communications and Information Sciences (KICS).

Wansu Lim received the B.S. degree in Electronic Engineering from Korea Aerospace University, Gyeonggi-do, Republic of Korea, in 2006. M.Sc. and Ph.D. degree in the area of optical and wireless communications from the Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea, in 2007 and 2010 respectively.

From 2010 to 2013, he was a Research Fellow at the University of Hertfordshire, U.K., and then a Postdoctoral Researcher (2013-2014) at the Institut National de la Recherche Scientifique (INRS), Quebec, Canada. Since September 2014, he has been a Professor in the Department of IT Convergence Engineering at Kumoh National Institute of Technology (KIT), Gumi, South Korea. His research interests include integrated optical/wireless access networks, device-to-device (D2D) communications, IoT, sensor networks, and artificial intelligence.

Dr. Lim is a member of the Institute of Electrical and Electronic Engineers (IEEE), the Korean Institute of Communications and Information Sciences (KICS), and IEEE Communications Society, and served as reviewer for several IEEE conferences and journals.