Blind channel equalization based on Complex-valued neural network and probability density fitting

Chouaib Farhati, Souhaila Fki, Abdeldjalil Aissa El Bey, Fatma Abdelkefi

Blind channel equalization based on Complex-valued neural network and probability density fitting

Chouaib Farhati [1] Souhaila Fki [1] Abdeldjali Aïssa El Bey [2] Fatma Abdelkefi [1]

(1)Supcom Lab-MEDIATRON, Carthage University, Ariana, Tunisie
(2)UMR CNRS 6285 Lab-STICC, IMT Atlantique, Brest, France, F-29238

chouaib.farhati@supcom.tn, feki.souhaila@gmail.com, abdeldjalil.aissaelbey@imt-atlantique.fr, fatma.abdelkefi@supcom.tn

***Abstract*—In this paper, we study blind equalization techniques to reduce the intersymbol interference (ISI) and we are particularly interested in equalizers based on probability density fitting (PDF). The PDF criterion was used with conventional linear equalizers. So we try in this paper to use this criterion in a nonlinear context using a neural network architecture. The network weights are updated by minimizing, at first, the stochastic quadratic distance, then the Multimodulus quadratic distance between the equalized PDF and some target distribution. Our approach shows a better performance in terms of mean square error (MSE) and symbol error rate (SER).**

***Index Terms*—blind equalization, neural network and probability density fitting.**

I. Introduction

In wireless communication, multipath channel introduces intersymbol interference (ISI). In the receiver side, it is difficult to estimate the received data without exploiting any prior processing. Equalization is one of the solutions used to reduce ISI. We are oriented towards blind equalization, because it allows a gain in flow and bandwidth given the absence of the periodic learning sequence. In this paper we are focused on probability density fitting (pdf) criterion especially the stochastic quadratic distance algorithm (SQD) [1] and the multimodulus stochastic quadratic distance algorithm(MSQD-ℓp) [2].

In the last decade, the neural network technique becomes a powerful tool to efficiently equalize channels. In the literature, several works have combined the neural network with the cost function of conventional equalizers to build blind neural equalizers like neural network constant modulus algorithm (NNCMA) [3] and neural network multimodulus algorithm (NNMMA) [4]. These algorithms have achieved better performance compared to their linear versions constant modulus algorithm (CMA) [5] and multimodulus algorithm (MMA) [6].

In [2], it has shown that SQD and MSQD-ℓp outperform CMA and MMA and it was demonstrated that the MSQD-ℓ1 algorithm converges close to the minimum mean squared error (MMSE) which is widely used as a benchmark. This encourages us to use these criterions with the neural network to improve the equalization performance.

In this paper we present two new blind neural equalization algorithms. The first is derived from SQD, where we combine the SQD cost function with the neural network and the second one is derived from MSQD-ℓp in a neural network context.

Several neural network architectures were used for blind equalization. We can mention for examples, the multilayer perceptron (MLP) or feedforward equalizer (FFE) [3], the feedforward with decision feedback equalizer (FFE-DF) [7], the recurrent neural network equalizer (RNNE) [8] [9] and variational autoencoders [10] [11]. Since the equalized signal is complex and to reduce the implementation complexity, in this paper, we have implemented a complex valued feed forward architecture. The latter is trained with complex backpropagation (CBP) [12] [13] [14] and an activation function dealing with the different constellation sizes.

This paper is organized as follows. In section II we propose the system model. In section III and IV we present respectively the first new algorithm wich is the neural network stochastic quadratic distance (NNSQD) and the second new algorithm wich is the neural network multimodulus stochastic quadratic distance (NNMSQD-ℓp). The simulations and results are presented in Section V. Conclusion is provided in section VI.

II. Signal and equalizer model

A. Signal model

Let $s(n)_{n \in Z}$ be a sequence of an independent and identically distributed (i.i.d) of complex symbols belonging to a particular modulation M-QAM and sent over a transmission channel $h = [h_0, h_1, ..., h_{L_{h-1}}]$ which we assumed in this paper finite impulse response (FIR). We denote by $b(n)$ an additive white Gaussian noise, $x(n)$ is the input of the equalizer, w are the synaptic weights of the neural network and take the role of equalizer impulse response and $y(n)$ is the equalized signal at time n such that.

$x(n) = \sum_{i=0}^{L_{h-1}} h_i s(n-i) + b(n)$ and

$y(n) = G(x, w)$,

where $x = [x(n), x(n-1), ..., x(n-Lw+1)]^T$, L_w is the size of the input neural network layer and G is the neural network total function.

The basic model of a transmission system using a neural equalizer as a filter is described in Fig.1

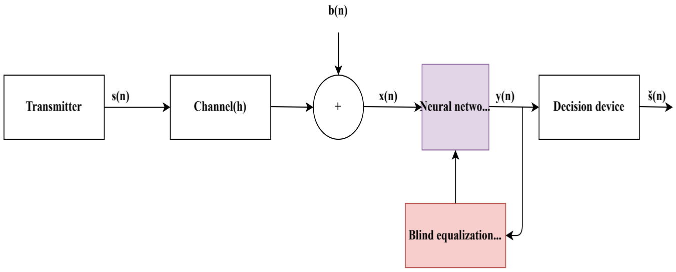

Fig. 1: Base band model of a transmission system with a neural network equalizer used as a filter.

B. Equalizer model

As mentioned in the above section, the neural network used for our equalizer is the complex-valued feed forward. whose structure is shown in Fig.2

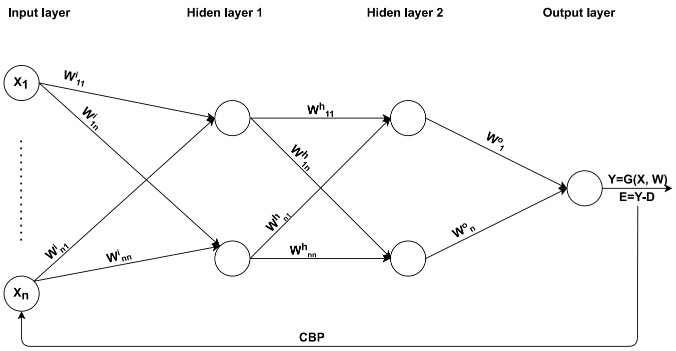

Fig. 2: A complex-valued feed forward structure.

The neuron structure of complex-valued feed forward is defined in Fig.3.

We assume N_k neurons in the k^{th} layer. Each neuron has an input named φ_j^k and an ouput named x_j^{k+1} that can be formulated as :

$$\varphi_j^k = \sum_{i=1}^{N_{k-1}} w_{ij}^k x_i^k + \theta_j^k, \tag{1}$$

$$x_j^{k+1} = f^k(\varphi_{j,R}^k) + j * f^k(\varphi_{j,I}^k), \tag{2}$$

where x_i^k, denotes the input of the layer, w_{ij}^k, is the weight between the i^{th} input and the j^{th} neurone in the k^{th} layer, θ_j^k denotes the bias and $f^k(.)$ is the activation function in the k^{th} layer.

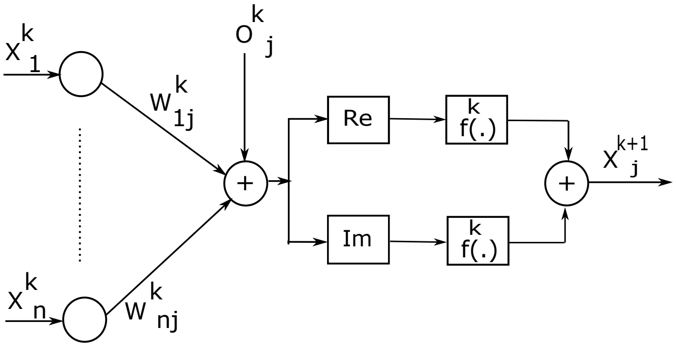

Fig. 3: Model of the j^{th} neurone in k^{th} layer in the complex-valued feed forward.

We can develop (1) as the following:

$$\varphi_{j,R}^k = \sum_{i=1}^{N_{k-1}} (w_{ij,R}^k x_{i,R}^k - w_{ij,I}^k x_{i,I}^k) + \theta_{j,R}^k \tag{3}$$

$$\varphi_{j,I}^k = \sum_{i=1}^{N_{k-1}} (w_{ij,R}^k x_{i,I}^k + w_{ij,I}^k x_{i,R}^k) + \theta_{j,I}^k \tag{4}$$

The complex feedforward equalizer designed during our contribution works in two main steps for each iteration. The first step is the forward pass from the input layer to the output layer. During this processing, the goal is to calculate the network output applying an activation function on the weighted sum in each layer. The second part is the complex back propagation (CBP) to update the synaptic weights according to our equalizer criterion.

III. Neural equalizers based on PDF criterion

The proposed equalizer is a mixture of the PDF criterion and the neural network. For this purpose, the cost function is based on the PDF during the CBP step. The two cost functions that will be used are SQD [1] and MSQD-$\ell 1$ [2]. In sequel, the CMA is also a widely known blind equalizer and is used as a reference in both conventional and neural modes [3]. The network has three layers : input, hidden and output layers. We assume along this paper that the activation function is :

$f(x) = x + \alpha * sin(\Pi * x)$.

This activation function is nonlinear under the effect of $sin()$ and the α parameter, which determines the degree of non linearity, is real positive. Moreover, this transfer function can deal with M-QAM signals of any constellation sizes [3].

A. Neural network stochastic quadratic distance equalizer(NNSQD)

SQD calculates the square distance between the probability density (PD) of the observations and the PD of the transmitted constellation. SQD has the following cost function [1]:

$$J_{SQD}(w) = \int_{-\propto}^{+\propto} (f_{Y^p}(z) - f_{S^p}(z))^2 dz, \tag{5}$$

where y is the output of the neural equalizer and $Y^p = |y(n)|^p$, S is the transmitted constellation and $S^p = |s(n)|^p.f_x(z)$ means the PD from x to z.

To estimate the PD of the current data, a Gaussian Kernel Parzen window was applied. Using this non parametric PD estimator with the previous L symbols, the PD at each discrete instant n is as follows :

$$\begin{aligned} f_{Y^p}(z) &= \frac{1}{L} \sum_{j=0}^{L-1} K_{\sigma_0}(z - |y(n-j)|^p), \\ f_{S^p}(z) &= \frac{1}{N_s} \sum_{k=1}^{N_s} K_{\sigma_0}(z - |s(k)|^p), \end{aligned} \tag{6}$$

where, N_s is the number of complex symbols in the constellation and K_{σ_0} is a Gaussian Kernel with zero mean and variance σ_0 which is denoted also Kernel width.
For $p = 2$ and $L = 1$, the cost function is

$$\begin{aligned} J_{SQD}(w) = &\frac{1}{N_s^2} \sum_{k=1}^{N_s} \sum_{l=1}^{N_s} K_{\sigma}(|s(l)|^2 - |s(k)|^2) \\ &- \frac{2}{N_s} \sum_{k=1}^{N_s} K_{\sigma}(|y(n)|^2 - |s(k)|^2), \end{aligned} \tag{7}$$

where $\sigma = \sqrt{2\sigma_0}$. Let μ be the learning rate, such that

$$w_{ij}^k(n+1) = w_{ij}^k(n) - \mu \frac{\partial J_{SQD}}{\partial w_{ij}^k(n)}. \tag{8}$$

The gradient is calculated as in [4]. We introduce two constants Q and δ_p^o to simplify the expression which are expressed as follows :

$$Q = \frac{4}{N_s} \sum_{k=1}^{N_s} \frac{1}{\sqrt{2\Pi}\sigma} e^{-\dfrac{(|y(n)|^2 - |s(k)|^2)^2}{2\sigma^2}} \frac{(|y(n)|^2 - |s(k)|^2)}{\sigma^2},$$

$$\delta_p^o = [f^o(\varphi_r^o) f^{o'}(\varphi_r^o) + j f^o(\varphi_i^o) f^{o'}(\varphi_i^o)]Q. \tag{9}$$

The weights of the output layer are updated as

$$w_j^o(n+1) = w_j^o(n) - \mu \delta_p^o I_{pj}^*, \tag{10}$$

where φ_r^o , φ_i^o , are the real and imaginary inputs of the output layer. I_{pj} is the output of the hidden layer. $f^o(\varphi_r^o)$ and $f^o(\varphi_i^o)$ are the activation functions of the output layer applied in the real and imaginary parts. $f^{o'}(\varphi_r^o)$ and $f^{o'}(\varphi_i^o)$ are the derivatives of the activation functions in output layer applied in the real and imaginary parts.

In the same way, to update the hidden layer weights, we applied the chain rule like in [4] and denote by δ_{pj}^h,

$$\delta_{pj}^h = f_j^{h'}(\varphi_r^h) Real(\delta_p^o w_j^{o^*}) + j f_j^{h'}(\varphi_i^h) Imag(\delta_p^o w_j^{o^*}). \tag{11}$$

Similarly, the weights of the hidden layer are updated as

$$w_{ji}^h(n+1) = w_{ji}^h(n) - \mu \delta_{pj}^h x_{pi}^*, \tag{12}$$

where φ_r^h , φ_i^h , are the real and imaginary inputs of the hidden layer and x_{pi} is the input of the network. $f_j^{h'}(\varphi_r^h)$ and $f_j^{h'}(\varphi_i^h)$ are derivatives of the activation functions of the hidden layer applied in the real and imaginary parts at the j^{th} neuron.

B. Neural network multimodulus SQD-ℓ1 equalizer (NNMSQD-ℓ1)

The decomposition of equalization criteria into an inphase component and a quadrature component is more efficient than processing two components together, with respect to a phase shift introduced by the transmission channel.
This approach will lead to the Multimodulus SQD-ℓp (MSQD-ℓp) criteria. More specifically, for p = 1, the MSQD-$\ell 1$, was introduced in [2] which has the following cost function :

$$\begin{aligned} J_{MSQD-\ell 1}(w) = &-\frac{1}{N_s} \sum_{k=1}^{N_s} K_{\sigma}(|y_r| - |s_r(k)|) \\ &- \frac{1}{N_s} \sum_{k=1}^{N_s} K_{\sigma}(|y_i| - |s_i(k)|), \end{aligned} \tag{13}$$

where N_s is the number of complex symbols in the constellation and K_{σ_0} is the same as in NNSQD. y_r and y_i and s_r and s_i are the real and imaginary parts of the equalized output and the emitted constellation, respectively.
We focus on the sequel on the derivation of gradient expression of the cost function. Using the same chain rule in [4] and for simplification purposes we introduce Q_r, Q_i and δ_p^o,

$$Q_r = \frac{1}{N_s} \sum_{k=1}^{N_s} \frac{1}{\sqrt{2\Pi}\sigma} e^{-\dfrac{(|y_r| - |s_r|)^2}{2\sigma^2}} \frac{(|y_r| - |s_r|)}{\sigma^2},$$

$$Q_i = \frac{1}{N_s} \sum_{k=1}^{N_s} \frac{1}{\sqrt{2\Pi}\sigma} e^{-\dfrac{(|y_i| - |s_i|)^2}{2\sigma^2}} \frac{(|y_i| - |s_i|)}{\sigma^2},$$

$$\delta_p^o = Q_r sign(f^o(\varphi_r^o)) f^{o'}(\varphi_r^o) + j Q_i sign(f^o(\varphi_i^o)) f^{o'}(\varphi_i^o). \tag{14}$$

The weights of the output layer are updated as

$$w_j^o(n+1) = w_j^o(n) - \mu \delta_p^o I_{pj}^* \tag{15}$$

where φ^o_r , φ^o_i , are the real and imaginary inputs of the output layer and I_{pj} is the output of the hidden layer. $f^o(\varphi^o_r)$ and $f^o(\varphi^o_i)$ are the activation functions of the output layer applied in the real and imaginary parts. $f^{o'}(\varphi^o_r)$ and $f^{o'}(\varphi^o_i)$ are the derivatives of the activation functions in output layer applied in the real and imaginary parts. Finally $sign$ is the sign function.

To update the hidden layer weights, we applied the chain rule like in [4] and simplify the expression by introducing δ^h_{pj},

$$\delta^h_{pj} = f^{h'}_j(\varphi^h_r)Real(\delta^o_p w^{o^*}_j) + jf^{h'}_j(\varphi^h_i)Imag(\delta^o_p w^{o^*}_j). \quad (16)$$

Similarly, the weights of the hidden layer are updated as

$$w^h_{ji}(n+1) = w^h_{ji}(n) - \mu\delta^h_{pj}x^*_{pi}, \quad (17)$$

where φ^h_r , φ^h_i , are the real and imaginary inputs of the hidden layer and x_{pi} is the input of the network. $f^{h'}_j(\varphi^h_r)$ and $f^{h'}_j(\varphi^h_i)$ are derivatives of the activation function of the hidden layer applied in the real and imaginary parts at the j^{th} neuron.

IV. Simulation results

We have simulated the three equalizers detailed in section III with a complex-valued feed forward neural network using three layers. The number of neurons in the input layer and the hidden layer depends on the modulation size. For 16-QAM and 64-QAM, we used 25 neurons [$L_i = 15$ (input), $L_h = 9$ (hidden), $L_o = 1$ (outout)] and for 256-QAM we used 101 neurones [$L_i = 40$ (input), $L_h = 60$ (hidden), $L_o = 1$ (output)].
The activation function is $f(x) = x + \alpha sin(\Pi x)$ and the value of α varies according to the equalizer type, modulation size, signal to noise ratio(SNR) and the layer order. In addition, we tested the equalizers performance when using a channel h_1 with real coefficients and one typical digital radio channel h_2 that was used in [2].

$$h_1(z) = 0.2258 + 0.5161z^{-1} + 0.6452z^{-2} - 0.5161z^{-3}$$

$$h_2(z) = (0.0410 + j0.0109) + (0.0495 + j0.0123)z^{-1} + (0.0672 + j0.0170)z^{-2} + (0.0919 + j0.0235)z^{-3} + (0.7920 + j0.1281)z^{-4} + (0.3960 + j0.0871)z^{-5} + (0.2715 + j0.0498)z^{-6} + (0.2291 + j0.0414)z^{-7} + (0.1287 + j0.0154)z^{-8} + (0.1032 + j0.0119)z^{-9}.$$

We have initialized the two matrices of synaptic weights $W^o[L,N]$ and $W^h[L]$ with small values to get $|w^{horo}_{RorI}| < 10^{-5}$. Except for $w^o[(L+1)/2,(N+1)/2]$ and $w^h[(L+1)/2]$ that were set to one.
For 16-QAM modulation, the length of the conventional equalizers L_w is set to 21. The learning rates were fixed to $\mu_{CMA} = 5e^{-6}$, $\mu_{SQD} = 5e^{-5}$ and $\mu_{MSQD-\ell 1} = 1e^{-4}$.

For the neuronal equalizers, we use the same value of α in the hidden and output layer, that was set to 0.1. The learning rates were fixed to $\mu_{NNCMA} = 3e^{-5}$, $\mu_{NNSQD} = 5e^{-5}$ and $\mu_{NNMSQD-\ell 1} = 1e^{-3}$.
For 64-QAM modulation, the length of the conventional equalizers L_w is set to 21. The learning rates were fixed to $\mu_{CMA} = 1.5e^{-7}$, $\mu_{SQD} = 6e^{-7}$ and $\mu_{MSQD-\ell 1} = 1e^{-4}$.

For the neuronal equalizers, we use the same value of α in the hidden and output layer, that was set to 0.1. The learning rates were fixed to $\mu_{NNCMA} = 8e^{-8}$, $\mu_{NNSQD} = 1e^{-7}$ and $\mu_{NNMSQD-\ell 1} = 2e^{-4}$.
For 256-QAM modulation, the length of the conventional equalizers L_w is set to 21. The learning rates were fixed to $\mu_{CMA} = 1e^{-8}$, $\mu_{SQD} = 5e^{-8}$ and $\mu_{MSQD-\ell 1} = 1e^{-5}$.

For the neuronal equalizers, we use the same value of α in the hidden and output layer, that was set to 0.1. The learning rates were fixed to $\mu_{NNCMA} = 2e^{-9}$, $\mu_{NNSQD} = 1e^{-8}$ and $\mu_{NNMSQD-\ell 1} = 8e^{-5}$.

The following three figures Fig.4, Fig.5 and Fig.6 represente the performance of the introduced equalizers in terms of MSE when using h_1 and h_2.

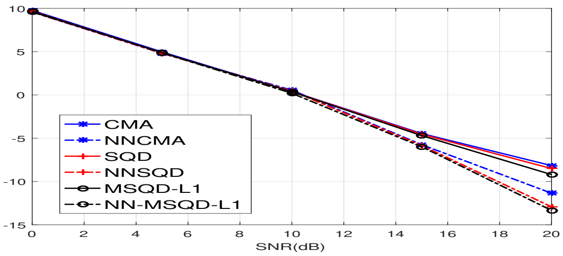

Fig. 4: MSE for 16-QAM and h_1.

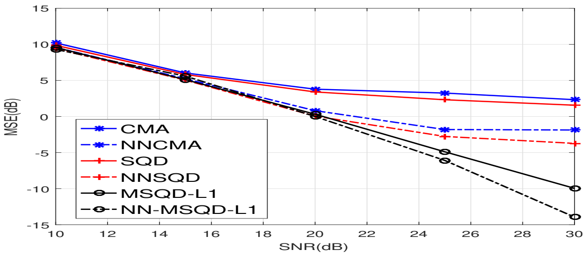

Fig. 5: MSE for 64-QAM and h_2.

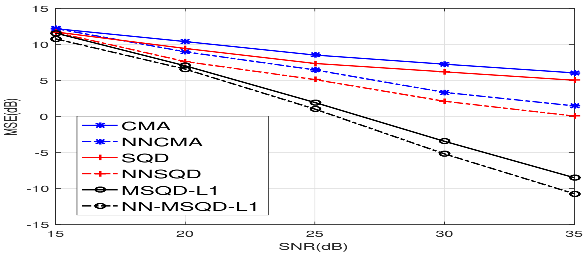

Fig. 6: MSE for 256-QAM and h_2.

We can notice according to the figures of MSE for different values of SNR that the NNMSQD-$\ell 1$ equalizer is the most performante in terms of MSE. The gain of MSE is more considerable when the value of SNR increases. This can be explained by the performance of the learning phase that increases when SNR increases. In addition, the NNMSQD-$\ell1$ equalizer is more efficient in terms of MSE than the MSQD-$\ell1$ which already achieves the performance of the non-blind equalizer MMSE [12].

Fig.7, Fig.8 and Fig.9 show the performance of the equalizers in terms of SER when using the same channels h_1 and h_2.

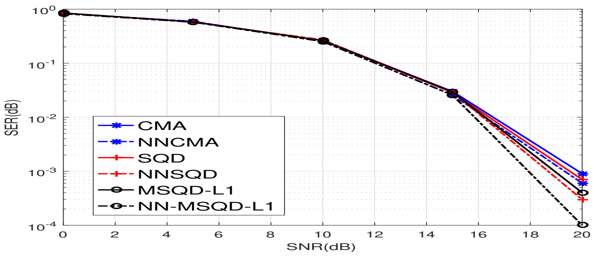

Fig. 7: SER for 16-QAM and h_1.

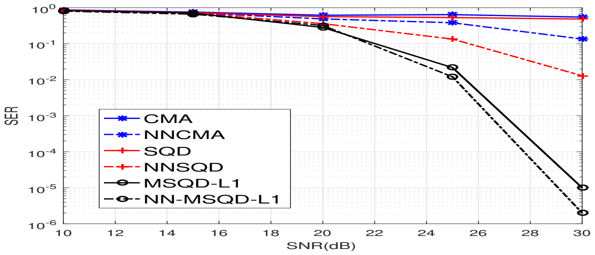

Fig. 8: SER for 64-QAM and h_2.

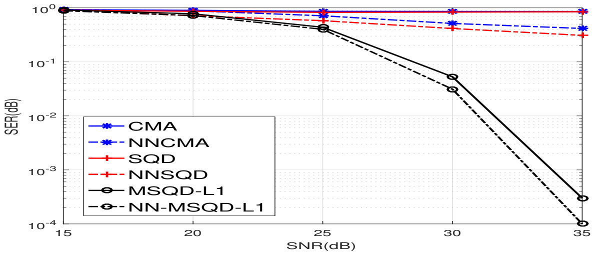

Fig. 9: SER for 256-QAM and h_2.

It is the same case here, we can see that NNMSQD-$\ell1$ has the best performance in terms of SER that confirms its performance in terms of MSE.

Computation burdens for NNSQD and NNMSQD-$\ell1$ is defined in Tab I.

Equalizer	**X**	**Exponents**
NNSQD	$3L_h(L_i + L_o) + 2L_h$	N_s
NNMSQD-$\ell1$	$3L_h(L_i + L_o) + 2L_h$	$2N_s$

TABLE I: Computational complexity

V. Conclusion

In this paper, we have proposed a new approach for blind equalization based on pdf fitting and neural network system. We have shown that it outperforms the conventionnal kernel blind equalization methods in terms of MSE and SER. As a perspective, we can generalize our algorithm with time varying channel. Deep learning has several applications in signal processing. It evolves quickly, it starts supervised, then unsupervised. Today, it becoms generic. This encourages us to open a track on the use of the generic deep learning model to create generic blind equalizers.

References

[1] Lázaro, Marcelino, Ignacio Santamaría, Deniz Erdogmus, Kenneth E. Hild, Carlos Pantaleón, and Jose C. Principe. "Stochastic blind equalization based on pdf fitting using parzen estimator." IEEE Transactions on Signal Processing 53, no. 2 (2005): 696-704.

[2] Fki, Souhaila, et al. "Blind equalization based on pdf fitting and convergence analysis." Signal Processing 101 (2014): 266-277.

[3] Wang, Donglin, and Dongming Wang. "Generalized derivation of neural network constant modulus algorithm for blind equalization." 2009 5th International Conference on Wireless Communications, Networking and Mobile Computing. IEEE, 2009.

[4] You, Cheolwoo, and Daesik Hong. "Nonlinear blind equalization schemes using complex-valued multilayer feedforward neural networks." IEEE transactions on neural networks 9.6 (1998): 1442-1455.

[5] Johnson, Richard, et al. "Blind equalization using the constant modulus criterion: A review." Proceedings of the IEEE 86.10 (1998): 1927-1950.

[6] Yang, Jian, J-J. Werner, and Guy Albert Dumont. "The multimodulus blind equalization and its generalized algorithms." IEEE Journal on selected areas in communications 20.5 (2002): 997-1015.

[7] Chemweno, Emmanuel K., Edward N. Ndung'u, and Heywood A. Ouma. "Adaptive Channel Equalization using Multilayer Perceptron Neural Networks with variable learning rate parameter." IOSR Journal of VLSI and Signal Processing 7.2 (2017): 34-40.

[8] Gagné, Christian, Ming Zeng, and Leslie A. Rusch. "Recurrent neural networks achieving MLSE performance for optical channel equalization." Optics Express 29.9 (2021): 13033-13047.

[9] Hu, Y., Zhao, L. and Hu, Y., 2019, June. Joint channel equalization and decoding with one recurrent neural network. In 2019 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB) (pp. 1-4). IEEE.

[10] Caciularu, A. and Burshtein, D., 2018, May. Blind channel equalization using variational autoencoders. In 2018 IEEE International Conference on Communications Workshops (ICC Workshops) (pp. 1-6). IEEE.

[11] Caciularu, A. and Burshtein, D., 2020. Unsupervised linear and nonlinear channel equalization and decoding using variational autoencoders. IEEE Transactions on Cognitive Communications and Networking, 6(3), pp.1003-1018.

[12] Leung, Henry, and Simon Haykin. "The complex backpropagation algorithm." IEEE Transactions on signal processing 39.9 (1991): 2101-2104.

[13] Benvenuto, Nevio, and Francesco Piazza. "On the complex backpropagation algorithm." IEEE Transactions on Signal Processing 40.4 (1992): 967-969.

[14] You, Cheolwoo, and Daesik Hong. "Adaptive equalization using the complex backpropagation algorithm." Proceedings of International Conference on Neural Networks (ICNN'96). Vol. 4. IEEE, 1996.

[15] Fki, Souhaila. Egalisation aveugle par méthodes à noyaux et techniques de classification automatique de modulations. Diss. Télécom Bretagne; Université de Bretagne Occidentale, 2015.

[16] Fki, Souhaila, Abdeldjalil Aïssa-El-Bey, and Thierry Chonavel. "Blind equalization and automatic modulation classification based on pdf fitting." 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2015.

[17] Fki, Souhaila, et al. "New criteria for blind equalization based on pdf fitting." 2014 IEEE 15th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC). IEEE, 2014.