



HAL
open science

Low-Complexity Adaptive Digital Predistortion with Meta-Learning based Neural Networks

Alexis Falempin, Rafik Zayani, Jean-Baptiste Dore, Emilio Calvanese Strinati

► **To cite this version:**

Alexis Falempin, Rafik Zayani, Jean-Baptiste Dore, Emilio Calvanese Strinati. Low-Complexity Adaptive Digital Predistortion with Meta-Learning based Neural Networks. 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), IEEE, Jan 2022, virtual event, United States. pp.453, 10.1109/CCNC49033.2022.9700529 . cea-03837858

HAL Id: cea-03837858

<https://cea.hal.science/cea-03837858v1>

Submitted on 3 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Low-Complexity Adaptive Digital Predistortion with Meta-Learning based Neural Networks

Alexis Falempin¹, Rafik Zayani², Jean-Baptiste Doré¹, Emilio Calvanese-Strinati¹

¹Univ. Grenoble Alpes, CEA, Leti, F-38000 Grenoble, France

²University of Carthage, LR-11/TIC-03 Innov'COM Laboratory, Sup'Com, Ariana 2083, Tunisia
alexis.falempin@cea.fr

Abstract—In this paper, we study a meta-learning based neural network (NN) model to enhance the energy-efficiency related to power amplification in wireless communication systems. Specifically, we introduce a low-complexity adaptive solution to perform digital predistortion (DPD) for power amplifiers (PAs) using neural networks. Thus, we design a dedicated NN architecture to derive predistortion functions using few neurons. Moreover, we develop a meta-learning training approach that allows better generalization and faster adaptation, over time-varying PAs, compared to classical DPD architectures. Thereby, we propose a new approach to realize an adaptive digital predistorter based on meta-learning using few samples for online calibration. A dedicated architecture allows to achieve low-complexity while meta-learning permits adapting to most parameters change in the system. Through the simulation results, we have shown that the developed meta-learning based NN DPD can offer a meta-trained DPD function, i.e., trained offline, that can provide satisfying performance for different PA models. Contrary to classical DPD architectures, the performance of our meta-trained DPD can be improved through only few gradient steps and over few samples during online calibration, achieving excellent performance with moderate complexity.

Index Terms—Communication systems, Physical layer, Energy-efficiency, Power amplifier, Digital pre-distortion, Machine learning, Neural networks, Meta Learning.

I. INTRODUCTION

In actual communication systems, high-power amplifiers (PAs) are primordial in the transmitter side so that the signal reaches its destination. Nevertheless, the PA is the most power-hungry component at the RF transmitter, about 80% of the total power consumed [1]. Thus, its power efficiency has to be significantly improved in order to improve the global energy-efficiency of radio frequency (RF) transmitters and reducing the carbon footprint. However, a RF PA exhibits nonlinear distortions when working close to its saturation level, where its power efficiency is high [2]. Indeed the PA presents amplitude-to-amplitude (AM/AM) and amplitude-to-phase (AM/PM) distortions, generating severe in-band and out-of-band (OOB) distortions on the wireless link affecting the performance at the receiver. Note that PAs present higher power efficiency when operating with lower back-off, i.e., closer to the saturation region. Hence, it is necessary to find a compromise between linearity and power efficiency.

Digital predistorsion (DPD) has been widely studied in literature [2] and has been shown to be the most promising PA linearization technique. It consists in adding a module before the PA such that the resulting system DPD and PA is linear.

However, estimating the DPD module, i.e., estimating the inverse characteristic of the PA is very challenging especially for time-varying PA. Besides, there has been a growing interest in the use of machine learning techniques for physical layer design of wireless systems [3] [4]. Indeed, neural networks (NNs) are suited to solve non linear problems. Hence, it is interesting to consider NNs for DPD. DPD based on NN has been investigated in many works such as [5] and [6]. Latter works present classic neural network architectures that do not take the complexity aspect into account. In [5], adaptive DPD is investigated but may require lot of data to calibrate the model efficiently. In [7], authors propose a solution based on NNs to avoid indirect learning algorithm. Although the good performance provided by these solutions, their complexity is still challenging when considering a time-varying PA. Indeed, the updating of the DPD function is performed on a per-PA model, and relearning is needed when the PA behavioral changes. Furthermore, the updating process requires sufficient amount of data and training time, which is not adequate with real-time communication systems.

In order to mitigate the inefficiency in terms of data and training time requirements, this paper introduces a meta-learning based NN DPD model which can provide satisfying performance whatever the PA characteristic. More importantly, the online calibration of the NN DPD can be efficiently carried out with reduced training data and computational complexity.

The main contributions of this work are twofold. First, we propose a custom architecture specifically dedicated to perform DPD in the polar domain, i.e. phase and amplitude domain. This custom design allows to achieve low-complexity. Second, we adopt meta-learning to provide an adaptive behaviour to our NN DPD relying on specific algorithm to improve NN generalization and fast calibration. We realize different tests of our solution on communication system integrating orthogonal frequency division multiplexing (OFDM), quadrature amplitude modulation (QAM) and a PA model derived from 3GPP specification [8].

The remainder of this paper is structured as follows. Sec. II-A presents the communication system model. Then, we present our NN DPD in Sec. II-B including NN architecture, training and inference stages. While Sec. III focuses on the chosen meta learning algorithm and its application, Sec. IV presents the simulation results and discussions. Finally, a conclusion is drawn in Sec. V with some future perspectives.

II. SYSTEM MODEL

A. OFDM signal and PA impairments

We consider a communication system integrating a QAM modulation, an OFDM transmitter, a DPD based on NN techniques and a PA. The communication system is pictured on Fig. 1.

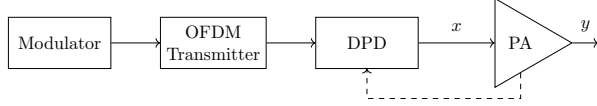


Fig. 1. System Model

The PA model is characterized by an amplitude distortion function denoted f_ρ and a phase shift function denoted f_Φ . The output characteristics are given by:

$$|y| = f_\rho(|x|) \quad (1)$$

$$\arg(y) = f_\Phi(|x|) + \arg(x) \quad (2)$$

where x and y are the input and output signals respectively. Assuming a PA derived from a 3GPP Rapp model for communication above 6GHz [8], the two functions are defined as follows:

$$f_\rho(u) = \frac{Gu}{\left(1 + \left|\frac{Gu}{V_{sat}}\right|^{2p}\right)^{\frac{1}{2p}}} \quad (3)$$

$$f_\Phi(u) = \frac{Au^q}{\left(1 + \left(\frac{u}{B}\right)^q\right)} \quad (4)$$

where G the gain in linear region, p the “knee” factor and V_{sat} the saturation voltage level. A , B and q are fitting parameters. Thereafter, we consider the following input back-off (IBO) definition :

$$IBO = \frac{P_{1dB,in}}{P_{avg,in}}, \quad (5)$$

where $P_{1dB,in}$ corresponds to the input power at the 1dB compression point and $P_{avg,in}$ the average input power.

Besides, we consider our model to be noise free.

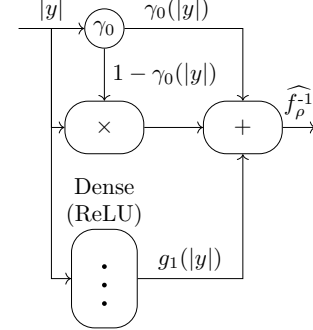
B. Conventional Neural Network DPD (CNNDPD)

In this section, we present our solution to perform the NN DPD function. It is based on the indirect learning architecture (ILA), which has been shown to provide better performance compared to the direct learning one. It is worth to mention that the studied scheme is different compared to the ones presented in the literature [5][7]. Indeed, instead of proposing a “blackbox” architecture, *i.e.* some fully connected hidden layers, we decide to design our neural network to specifically tackle amplitude and phase impairments separately through a polar decomposition of the signal. According to [9], tackling AM/AM and AM/PM distortions gives better results which motivates this choice. It results in a specific architecture.

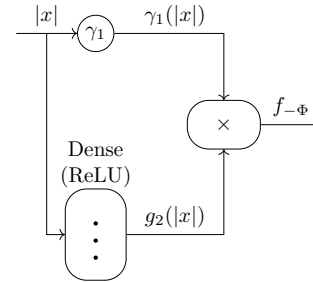
The CNNDPD is composed of two neural networks. Each neural network represents a function correcting respectively

the amplitude distortion (AM/AM) and the phase distortion (AM/PM).

The architecture of both neural networks is presented in Fig. 2. It must be emphasized that these neural networks respectively use amplitude and phase information of the signal.



(a) Predistorsion function for AM/AM distortion



(b) Predistorsion function for AM/PM distortion

Fig. 2. Architecture of the CNNDPD

1) *Neural Network for AM/AM DPD:* In this paragraph, we present the architecture of the neural network allowing to invert the PA AM/AM characteristic represented on Fig. 2a. First, it can be noticed that this architecture is fully customized in order to resolve the specific issue of inverting the PA AM/AM characteristic. Based on the system model presented in Sec. II, the NN presented here takes the output amplitude of the PA and predicts the input amplitude. This NN is optimized to find the function \widehat{f}_ρ^{-1} such as,

$$(f_\rho \circ \widehat{f}_\rho^{-1})(|x|) = G|x|, \quad (6)$$

$$\widehat{f}_\rho^{-1}(u) = g_0(u) + g_1(u),$$

$$g_0(u) = \gamma_0(u) + (1 - \gamma_0(u))u,$$

$$\gamma_0(u) = \left(1 + e^{-\alpha(u - \omega_\rho)}\right)^{-1},$$

$$g_1(u) = \sum_{j=1}^{N_n^\rho} \omega_j^\rho [\text{ReLU}(\mathbf{W}_\rho u + \mathbf{b}_\rho)]_j,$$

where $\alpha, \omega_\rho, \omega_j^\rho \in \mathbb{R}$, $\mathbf{W}_\rho \in \mathbb{R}^{1 \times N_n^\rho}$ and $\mathbf{b}_\rho \in \mathbb{R}^{1 \times N_n^\rho}$ are optimized during the learning phase. N_n^ρ the number of neurons. The function g_0 allows to model the invert AM/AM characteristic and the function g_1 allows to refine the DPD

function estimated by g_0 . ReLU function is defined by $f(x_i) = \max(0, x_i)$, $x_i = [\mathbf{x}]_i$.

2) *Neural Network for AM/PM DPD*: We present here the NN architecture used to correct phase shift due to the PA. The NN takes the input amplitude and predict the opposite of the phase shift. The architecture is different from the NN dedicated to invert the AM/AM characteristic because here we need to find an opposite which is likely simpler. Hence, fewer operations are required resulting in lower complexity. This NN is optimized to find the function $f_{-\Phi}$ such as,

$$\begin{aligned} f_{\Phi}(|x|) + f_{-\Phi}(|x|) &= 0, \\ f_{-\Phi}(u) &= \gamma_1(u)g_2(u), \\ \gamma_1(u) &= \left(1 + e^{-\beta(u-\omega_{\Phi})}\right)^{-1}, \\ g_2(u) &= \sum_{j=1}^{N_n^{\Phi}} \omega_j^{\Phi} \text{ReLU}[(\mathbf{W}_{\Phi}u + \mathbf{b}_{\Phi})]_j, \end{aligned} \quad (7)$$

where $\beta, \omega_{\Phi}, \omega_j^{\Phi} \in \mathbb{R}$, $\mathbf{W}_{\Phi} \in \mathbb{R}^{1 \times N_n^{\Phi}}$ and $\mathbf{b}_{\Phi} \in \mathbb{R}^{1 \times N_n^{\Phi}}$ are optimized during the training phase. N_n^{Φ} the number of neurons. In order to acquire the phase shift $f_{\Phi}(|x|)$, we realize a simple conjugate operation,

$$\begin{aligned} z &= x^*y, \\ z &= |x| \cdot |y| e^{i(\arg(y) - \arg(x))}, \\ \arg(z) &= f_{\Phi}(|x|), \end{aligned} \quad (8)$$

3) *Training*: The first approach to train our solution is by using Indirect Learning Algorithm (ILA). It consists in deriving a postdistorter and placing it before the PA to perform DPD. Learning is performed by optimizing a mean squared error (MSE) loss function using an Adam [10] optimizer. This training approach is known as “conventional” because we realize a simple gradient descent to update the NN parameters.

III. META LEARNING

In this section, we present the use of meta learning to enable adaptive DPD.

A. Limitations of conventional approach

In this paragraph, we present the limitations of the approach presented above. The goal of our work is to conceive an adaptive DPD solution with low-complexity. We also seek small data usage in order to minimize global latency of the system due to online calibration. Thereafter, we consider an adaptive scenario where the parameter p of the PA model Eq. (3) is varying. Fig. 3 presents the EVM performance considering that p is varying and an $I\text{BO}_{dB} = 6\text{dB}$. The x axis corresponds to a range of DPD trained for $p = 0.7 + 0.1k$, $k \in [0, 8]$ and the y axis corresponds to the value of p during performance test with $p = 0.7 + 0.01k$, $k \in [0, 80]$. Parameters of this simulation can be found in Table I.

We can observe that the NN achieves its optimal on the diagonal which correspond to a trained p for the same inferred

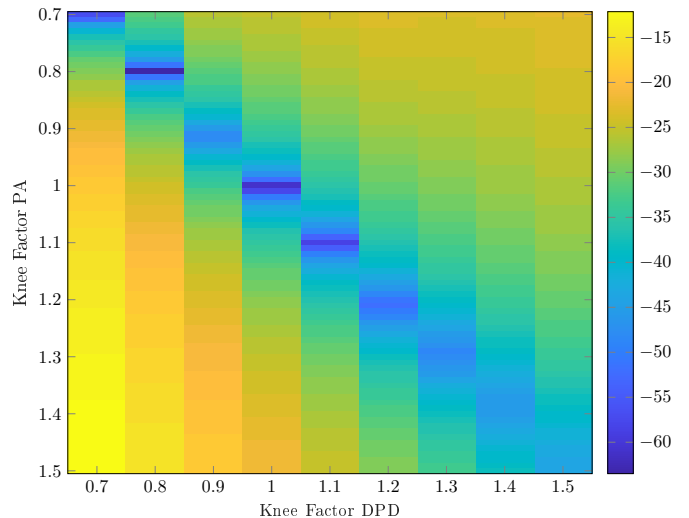


Fig. 3. EVM performance in [dB] with $0.7 \leq p \leq 1.5$

p . Moreover, we can underline the fact that the performance is severely degraded when a trained value of p is inferred on another value. Hence, the model is not able to adapt to a model variation. However, it could be envisaged to retrain the model online but it would consume a lot of symbols for online training phase. Thus, in the following section, we investigate meta learning as a solution to improve the robustness of our NN DPD to misadpatations.

B. Meta learning at glance

Meta learning can be seen as a “learn to learn” method meaning that we improve a global learning algorithm using multiple trainings [11]. On the contrary, “conventional” learning improves predictions using a single training on batches of data. Concretely, conventional learning is said to be task specific meaning it will learn exclusively for one configuration. This can be problematic if we have a system with multiples configurations because adaptation will be slow and data consuming. On the other hand, meta learning works by learning on multiples configurations, called tasks, such that it can quickly train on a configuration corresponding to the task. The objective is to quickly train or adapt a model from few data.

In wireless communication, systems configuration often need to adapt due environment change or limitations. Therefore, meta learning usage can help reducing data cost and training complexity. The use of meta learning in wireless communications is relatively new and few works are related to its usage. In [12], authors claim that meta learning can reduce training overhead and complexity by minimizing pilot symbols usage. Related works [13] and [14] introduce meta learning usage for end-to-end learning and few pilots demodulation. These works show that meta learning significantly reduce data usage and training time. However, to best author’s knowledge, there is no investigation on the use of meta learning for

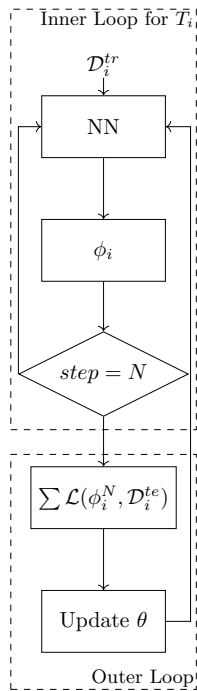


Fig. 4. MAML algorithm

adaptive DPD. Hence, in this section, we investigate meta learning to perform DPD with few data.

C. Model-Agnostic Meta Learning (MAML)

In this paragraph, we present the MAML algorithm [15] which is a meta learning algorithm enabling generalization and fast adaptation of neural networks.

The goal of this algorithm is to find the best weights to initialize the neural network for later inference on a dataset. Given that objective, this algorithm could be seen as an optimized initializer. In “conventional” learning, we often randomly initialize neural network weights which can slow the convergence and lead to poor generalization. MAML algorithm can be divided in two major steps, finding the best weights to initialize our NN model and converging to adapted weights.

1) *Learning weights initialization*: This step represents the core of the MAML algorithm. Indeed, the goal of MAML is to quickly train a NN model for a specific configuration. This objective is mainly achieved by finding the weights θ to initialize our NN model. This stage is called meta training.

The weights θ are inferred using data from multiple configurations/tasks \mathcal{T}_i , belonging to the same distribution $p(\mathcal{T}_i)$. A task \mathcal{T}_i is characterized by two datasets a training dataset \mathcal{D}_i^{tr} and a test dataset \mathcal{D}_i^{te} . More concretely, a task depends on the variation of one or multiples parameters. As a toy example, we want to derive the function $y = A \sin x + \varphi$ where A and φ are respectively the amplitude and the phase. Using MAML, we would learn on multiples tasks parametrized by A and φ . Each task consists in retrieving y from x with different amplitude and phase.

MAML algorithm is represented on Fig. 4. The first step consists in sampling a batch of tasks \mathcal{T}_i from $p(\mathcal{T}_i)$. Each task is composed of datasets \mathcal{D}_i^{tr} and \mathcal{D}_i^{te} . Then, we derive optimal weights ϕ_i foreach task \mathcal{T}_i using gradient descent. If we consider one gradient step, ϕ_i is given by,

$$\phi_i = \theta - \alpha \nabla_{\theta} \mathcal{L}(\theta, \mathcal{D}_i^{tr}), \quad (9)$$

where α is the inner learning rate. This step is called meta-update or inner loop and can be seen as a “conventional” learning on each task \mathcal{T}_i .

Next, from the optimal weights ϕ_i , MAML infers the weights θ solving the following optimization problem:

$$\min_{\theta} \sum_i \mathcal{L}(\phi_i, \mathcal{D}_i^{te}). \quad (10)$$

This problem is solved by performing a gradient descent leading to,

$$\theta = \theta - \beta \nabla_{\theta} \sum_i \mathcal{L}(\phi_i, \mathcal{D}_i^{te}), \quad (11)$$

with β the outer learning rate. This phase is called the meta-optimization or the outer loop update.

2) *Learning adapted weights*: Once general weights θ are inferred, an adaptation stage must be done to find optimal weights ϕ . This step is straightforward because it represents a “conventional” learning using previously inferred θ weights as initialization for our NN instead of random initialization. Thus, we get ϕ using gradient descent,

$$\phi^k = \phi^{k-1} - \nabla_{\phi^{k-1}} \mathcal{L}(\phi^{k-1}, \mathcal{D}^{tr}), \quad (12)$$

with $k = 1, \dots, N$ the number of gradient steps, $\phi^0 = \theta$ and \mathcal{D}^{tr} the samples to update the model to optimal solution.

To fasten this stage, one may consider using an Adam optimizer [10] to find the optimal weights.

3) *Improvements to MAML algorithm*: First, MAML can be improved to fasten and ensure convergence using loss optimizers instead of regular Stochastic Gradient Descent (SGD) as presented in Eq. (9) and Eq. (11). Some optimizers such as Adam [10] or Rectified Adam [16] greatly improve the convergence by adapting the learning rate of the algorithm.

Then, some core optimizations about the MAML algorithm could be made to improve its convergence. In [17], authors have investigated some improvements that can be done in order to enhance generalization and convergence. Specifically, instead of optimizing the outer loss after doing all the inner steps, it can be envisaged to optimize the outer loss after every inner step. Doing so allows to improve gradient stability of MAML according to the authors. Other improvements made in this paper are not that much relevant for our case because we use small architectures and benefits would be lesser.

D. MAML based Adaptive DPD

Regarding the meta learning techniques and state-of-the-art, MAML can help to perform an adaptive DPD using less data. To comply with the MAML algorithm, tasks must be chosen carefully. Thus, we propose an adaptive DPD to the variation of parameter p in Eq. (3) to illustrate the benefits of using meta learning. Thereafter, we consider a task \mathcal{T}_i being characterized by a value of the parameter p denoted p_i . Thus, a task consist in deriving $|x_i|$ from $|y_i|$ – refer to Sec. II – leading to derive optimal DPD for parameter p_i .

IV. NUMERICAL SIMULATIONS

In this section, we present numerical results to underline the benefits of meta learning approach compared to “conventional” learning. Thus, we first detail the “conventional” learning performance and limitations in order to promote meta learning algorithm.

We begin considering a varying PA model version of the characteristics presented in Sec. II. To demonstrate the benefits of meta learning algorithm for DPD, we compare “conventional” learning and meta learning for adaptation to the parameter p . Thereafter, we consider that $p \in [0.7, 1.5]$. This choice allows to cover a range of PA with few and lots of non linearities to bring diversity to the simulations. For both “conventional” learning and meta learning algorithms, we consider having working on the system model presented in Sec. II with a 64-QAM, a FFT size of 1024 for OFDM modulation.

A. Conventional learning

In this paragraph, we present the performance of “conventional” learning on static and varying PA model based on Eq. (3) and Eq. (4). For simulation purpose, we consider that parameters of the PA model are fixed according to [8], *i.e.* $p = 1.1$, $V_{sat} = 1.9V$, $G = 16$, $A = -345$, $B = 0.17$ and $q = 4$. The PA characteristics, AM/AM and AM/PM distortions are presented on Fig. 5.

Fig. 6 presents the Error Vector Magnitude (EVM) in function of the Input Back-Off (IBO) using a PA which presents AM/AM and AM/PM distortions. The “Limit” curve corresponds to a PA linear until its saturation characterized by $\min(G|x|, V_{sat})$. The performance of our solution is almost optimal using only 10 neurons which justify the low-complexity aspect. It must be emphasized that such a solution requires a large amount of data for a single state of our PA, about 10^5 OFDM symbols which represents a cumbersome database. The parameters of the CNNDPD are given in Table I.

Sec. III-A presents the limitations of CNNDPD regarding an adaptive scenario where p is varying. The main issue encountered is that the model is not suited to a PA model variation. One may consider using transfer learning and online retraining but this solution is not viable for high date rate wireless communications because we need to much gradient steps to obtain well performance.

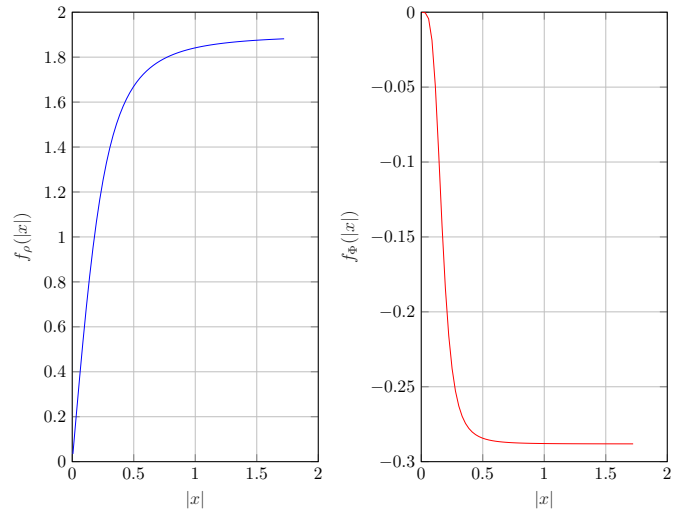


Fig. 5. AM/AM and AM/PM characteristics of Rapp PA [8]

TABLE I
IMPLEMENTATION PARAMETERS OF CNNDPD

Parameters	Values
N_n^ρ	8
N_n^ϕ	4
Batch size	128
Total data	10^5
Epochs	50
Optimizer	Adam
Learning rate	0.001
Loss function	Mean Squared Error
Activation function	ReLU

B. Meta Learning: From training to numerical results

We give here some detailed practical implementation aspects regarding training and inference of our solution to perform efficient DPD. First, meta learning parameters are summarized in Table II. Next, about the training phase, it must be underlined that the range of tasks must be chosen carefully. Indeed, increasing the parameter p will lead to a more linear model of the PA. Hence, having a range of to many linear PAs will decrease the MAML performance for correcting non linearities. Thus, we choose to train our model with $p \in [0.7, 1.5]$. This range ensures to have sufficient non linearities to perform DPD. Moreover, the number of tasks shall also be discussed. In our case, we are trying to minimize the required amount of data to train and infer the algorithm. We have noticed that a total number of 17 tasks is sufficient to have optimal convergence. Increasing this number will not offer a significant gain in terms of performance and will also increase training time.

Besides, building the database is a crucial step to ensure convergence and performance of the solution. Concerning MAML usage, we must build two datasets, \mathcal{D}_i^{tr} and \mathcal{D}_i^{te} with the same amount of data – refer to Sec. III. Both datasets are

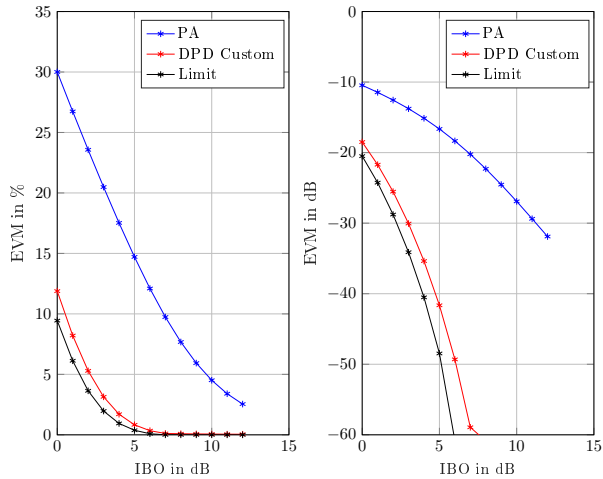


Fig. 6. EVM performance in [dB] using “conventional” learning

TABLE II
META LEARNING PARAMETERS

Parameters	Values
Tasks number	17
Inner steps	3
Total data	51000
Epochs	250
Optimizer	Rectified Adam
α	10^{-3}
β	5.10^{-4}
Loss function	Mean Squared Error
Activation function	ReLU

composed of PA output and input OFDM symbols amplitudes. We then have $\mathcal{D}_i^{tr} = (|\mathbf{y}_{tr}|, |\mathbf{x}_{tr}|)$ and $\mathcal{D}_i^{te} = (|\mathbf{y}_{te}|, |\mathbf{x}_{te}|)$ with $\mathbf{x}_{tr}, \mathbf{y}_{tr} \in \mathbb{R}^{1 \times B_{tr}}$ and $\mathbf{x}_{te}, \mathbf{y}_{te} \in \mathbb{R}^{1 \times B_{te}}$. B_{tr} and B_{te} denote the datasets sizes. We consider having for our training $B_{tr} = B_{te} = B$. Choosing the value of B is important for well generalization of the model. However, choosing a high number of data would significantly impact convergence due to the gradient iteration over multiple tasks. Therefore, we consider having a training batch with $B = 1000k, k \in [1, 10]$. Empirically, we found that using $2000 < B < 5000$ gives the best convergence considering that we use multiple inner gradients steps k , *i.e.* the number of inner loop updates in MAML. Thereby, for each inner step k , we perform a gradient descent over 1000 OFDM symbols for our batch B . Thus, using $B = 3000$, we would have 3 inner steps. Using different symbols for each inner step allow to bring more diversity and better convergence. For each task, choosing $B < 1000$ OFDM symbols may cause an issue because the statistical distribution of the data will not fully cover the PA model leading to non optimal DPD.

Finally, inference stage is also important. We initialize our CNNDPD with the learned weights in MAML phase. For any variation of the PA model, an online training will be performed with small gradient updates to improve system latency. In our

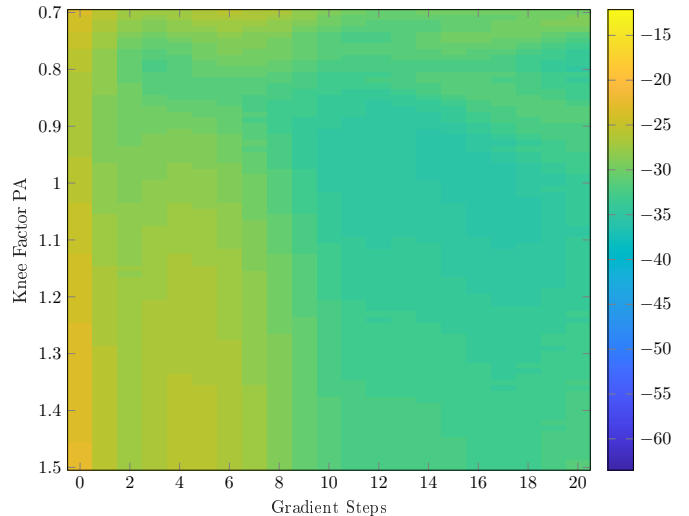


Fig. 7. EVM performance in [dB] with $0.7 \leq p \leq 1.5$

case, a single batch of 1000 OFDM symbols is used perform to perform online learning for inference.

Fig. 7 presents the performance of MAML based DPD over a range of PA with $p = 0.7 + 0.01k, k \in [0, 80]$ in function of gradient steps. First, we can observe that without retraining, *i.e.* gradient steps equals 0, MAML achieve an average of -25dB which is fair compared to conventional learning where we can reach only -10dB in some cases.

Increasing the gradient steps will notably improve the performance. Using 13 gradient steps, we achieve an average of -35dB EVM which is a descent performance considering the wide range of PAs. It shall be reminded that we only used 17 tasks for training. However, here inference is conducted over 80 PA models. It shows that MAML can adapt to unseen tasks but still belonging to the same training task distribution.

C. Discussion

The conducted simulations allow us to state over the benefits of meta learning use for adaptive DPD issue. Considering fixed PA parameters, the “conventional” learning will achieve near optimal performance, around -60dB EVM for an $IBO_{dB} = 6\text{dB}$. However, if we consider adapting our DPD to another PA model, meta learning outperforms the conventional learning in terms of data usage and gradients steps. Indeed, with only 13 gradient steps we can have up to 20dB gain compared to “conventional” learning on the whole range of PA, except when the DPD is inferred for the same trained p value.

Thus, it can be said that MAML brings advantages such as generalization, few data usage and fast convergence to propose an adaptive DPD. However, we still have a performance gap due to the generalized meta model. An expert model will always be better for a specific task but meta learning is better for multiple tasks problems.

V. CONCLUSION

In this paper, we propose a new approach to perform both low-complexity and adaptive DPD based on neural networks

to cancel non-linear effects caused by PAs in a wireless transmission chain. To achieve these goals, we first designed two custom neural networks that are specially dedicated to correct respectively AM/AM and AM/PM distortions of PAs. This particular design allows to achieve low-complexity, about 12 neurons for both NNs. The benefit of this architecture is also to enable fully parallelized operations which drastically reduce computational time compared to classic architecture with many hidden layers. In addition, to answer the adaptive concern of the DPD, we propose to use a meta learning approach for training and inferring our low-complexity neural networks. Meta learning and specifically MAML allow to find the best parameters initialization for the NNs. Based on learning over multiple tasks, we are able to find weights close to the optimal achievable DPD. Next, an online retraining with few samples and few training steps permits achieving fair performance compared to conventional learning. Numerical results show that meta learning gives descent performance even for cases unseen during training stage. This implies that our solution is able to adapt easily to a variation of the PA.

Thus, we truly believe that our proposed solution brings the key to provide low-complexity, adaptive and efficient DPD for PAs. Further investigation is on going to test our approach on a more realistic system model including a real power amplifier, analog-to-digital and digital-to-analog converters. We intend to correct effects of all these components both in terms of performance and spectral efficiency.

REFERENCES

- [1] G. Auer, O. Blume, and V. Giannini, "Energy efficiency analysis of the reference systems, areas of improvements and target breakdown," pp. 1–68, 12 2010.
- [2] S. Cripps, *RF Power Amplifiers for Wireless Communications, Second Edition*. Artech House, 2006.
- [3] T. O'Shea and J. Hoydis, "An Introduction to Deep Learning for the Physical Layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.
- [4] H. Huang, S. Guo, G. Gui, Z. Yang, J. Zhang, H. Sari, and F. Adachi, "Deep Learning for Physical-Layer 5G Wireless Techniques: Opportunities, Challenges and Solutions," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 214–222, 2020.
- [5] R. Zayani, R. Bouallegue, and D. Roviras, "An adaptive neural network pre-distorter for non stationary HPA in OFDM systems," in *2007 15th European Signal Processing Conference, 2007*, pp. 1352–1356.
- [6] A. S. Ciminski, "Recurrent neural networks usefulness in digital pre-distortion of power amplifiers," in *15th International Conference on Microwaves, Radar and Wireless Communications (IEEE Cat. No.04EX824)*, vol. 1, 2004, pp. 249–252 Vol.1.
- [7] C. Tarver, L. Jiang, A. Sefidi, and J. R. Cavallaro, "Neural Network DPD via Backpropagation through a Neural Network Model of the PA," in *2019 53rd Asilomar Conference on Signals, Systems, and Computers, 2019*, pp. 358–362.
- [8] Nokia, "Realistic power amplifier model for the New Radio evaluation," 3GPP TSG-RAN WG4, Tech. Rep., 2016.
- [9] R. Zayani, Y. Medjahdi, H. Bouhadda, H. Shaiek, D. Roviras, and R. Bouallegue, "Adaptive predistortion techniques for non-linearly amplified fbmc-oqam signals," in *2014 IEEE 79th Vehicular Technology Conference (VTC Spring), 2014*, pp. 1–5.
- [10] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2017.
- [11] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-Learning in Neural Networks: A Survey," 2020.
- [12] O. Simeone, S. Park, and J. Kang, "From Learning to Meta-Learning: Reduced Training Overhead and Complexity for Communication Systems," *CoRR*, vol. abs/2001.01227, 2020.
- [13] S. Park, O. Simeone, and J. Kang, "Meta-Learning to Communicate: Fast End-to-End Training for Fading Channels," 2019.
- [14] S. Park, H. Jang, O. Simeone, and J. Kang, "Learning to Demodulate from Few Pilots via Offline and Online Meta-Learning," 2020.
- [15] C. Finn, P. Abbeel, and S. Levine, "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks," *CoRR*, vol. abs/1703.03400, 2017.
- [16] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the Variance of the Adaptive Learning Rate and Beyond," *CoRR*, vol. abs/1908.03265, 2019.
- [17] A. Antoniou, H. Edwards, and A. J. Storkey, "How to train your MAML," *CoRR*, vol. abs/1810.09502, 2018.