EPJ Web of Conferences **380**, 01008 (2026) https://doi.org/10.1051/epjconf/202638001008
ICISCT'25

Comparative Analysis of Neural Network Architectures for Digital Predistortion in Power Amplifier Linearization

Hafsa Laakouri[1,*], *Mostapha* Ouadefli[1,**], *Abdelwahed* Tribak[1,***], *Mohamed* Et-tolba[1,****], and *Jaouad* Terhzaz[2,†] and *Tomas* Fernandez Ibanez[3,‡]

[1]National Institute of Postes and Telecomunications (INPT) Rabat, Morocco
[2]Centre regional des métiers de l'éducation et de formation (CRMEF) Casablanca, Morocco
[3]Universidad de Cantabria (UNICAN), Santander, Spain

Abstract. In this study, we examine and contrast the effectiveness of different artificial neural network (ANN) topologies for power amplifier (PA) digital predistortion (DPD). In particular, we investigate long short-term memory (LSTM) networks, gated recurrent units (GRU), recurrent neural networks (RNN), convolutional neural networks (CNN), and fully connected neural networks (DNN). For training and assessment, a dataset comprising measured input and output signals from a commercial NXP Doherty PA working in the 3.6–3.8 GHz region with a 16-QAM OFDM signal is utilised. Normalised mean squared error (NMSE), adjacent channel power ratio (ACPR), and model complexity are used to evaluate the models. Simulation results show that while CNNs offer a favorable trade-off between linearization performance and model complexity, GRU and LSTM architectures achieve the best overall NMSE and ACPR improvements, albeit with a higher number of parameters. Power spectral density and AM/AM characteristic analyses further confirm the superior linearization performance achieved using recurrent gated structures.
Index Terms: Digital predistortion (DPD), Power amplifier linearization, Neural Network, FFNN, CNN, RNN, GRU, LSTM.

1 Introduction

Stricter specifications for transmitter linearity and spectrum efficiency have resulted about by the quick growth of wireless communication systems, particularly with the implementation of 5G networks. Power amplifiers (PAs), as inherently nonlinear devices, introduce amplitude and phase distortions that degrade signal quality and cause spectral regrowth, leading to adjacent channel interference [1]. To mitigate these effects and comply with spectral emission masks, digital predistortion (DPD) techniques are widely employed in modern RF transmitters.

*e-mail: laakouri.hafsa@doctorant.inpt.ac.ma
**e-mail: mostapha.ouadefli@doctorant.inpt.ac.ma
***e-mail: tribak@inpt.ac.ma
****e-mail: ettolba@inpt.ac.ma
†e-mail: terhzazj@yahoo.fr
‡e-mail: tomas.fernandez@unican.es

Traditional DPD approaches, such as memory polynomial models [2], often struggle to accurately capture the highly nonlinear and memory-dependent behavior of wideband PAs operating under complex modulation schemes. As communication systems demand wider bandwidths and higher peakto-average power ratios, more advanced nonlinear modeling techniques have become necessary.

Recently, machine learning (ML) and artificial neural networks (ANNs) have emerged as promising tools for PA modeling and predistortion [3]. Fully connected deep neural networks (DNNs) have been proposed to approximate complex PA behaviors with good generalization properties [4]. Convolutional neural networks (CNNs) have been introduced to exploit local temporal features in PA input output relationships ,[5]. Similarly, recurrent neural networks (RNNs) have been applied to model dynamic nonlinearities owing to their internal memory mechanisms [6]. Gated architectures such as long short-term memory (LSTM) networks [7] and gated recurrent units (GRUs) [8] have demonstrated superior capabilities in learning long-term dependencies while mitigating the vanishing gradient problem, making them particularly well-suited for dynamic DPD tasks.

In this paper,we compare several ANN topologies used in DPD for power amplifier linearization. Specifically, we investigate the performance of DNNs, CNNs, RNNs, GRUs, and LSTMs. Each model is trained using a measured dataset obtained from an NXP Doherty PA excited with a 140 MHz 5G-like OFDM signal employing 16-QAM modulation[9].The models are assessed using model complexity, adjacent channel power ratio (ACPR), and normalized mean squared error (NMSE). Additionally, comparisons of AM/AM characteristics and power spectral density (PSD) are provided to show how various network configurations affect the linearization of PA. The remainder of this document is structured as : The neural network architectures and the models' mathematical foundation are displayed in Section 2. The simulation results are described in detail and a comparative analysis is given in Section 3. The paper is concluded finally in Section 4 .

2 Existing NN-Based DPD Models

2.1 Fully connected neural networks (FCNN)

An FCNN built with the Keras Sequential API is the first model being studied. Three hidden layers, each consisting of thirty neurones with exponential linear unit (ELU) activations, are used by the network to analyse a 20-dimensional input vector that represents the current and previous samples of the PA input signal. By reducing vanishing-gradient problems and enabling the network to generate negative outputs—which can be useful for modelling symmetric distortions—ELU activations are selected to speed up learning. Following the concealed layers is a final dense layer with two linear outputs that generates the predistorted signal's imaginary and real components.

This straightforward architecture serves as a baseline for comparing more specialized network types (CNN, RNN, LSTM, GRU) in terms of linearization performance and computational complexity. The input data used to train all the NN-architectures is given by:

$$X(n) = \begin{pmatrix} I_{in}(n-M) & I_{in}(n-M+1) & \dots & I_{in}(n) \\ Q_{in}(n-M) & Q_{in}(n-M+1) & \dots & Q_{in}(n) \\ |x(n-M)| & |x(n-M+1)| & \dots & |x(n)| \\ |x(n-M)|^2 & |x(n-M+1)|^2 & \dots & |x(n)|^2 \\ \vdots & \vdots & \vdots & \vdots \\ |x(n-M)|^K & |x(n-M+1)|^K & \dots & |x(n)|^K \end{pmatrix}$$

where $I(n)$ and $Q(n)$ denote the real and imaginary parts of the complex signal $X(n)$, and $|x(n)|$ stands for its amplitude. Here, M and K indicate the memory depth and non-linearity order , respectively.

2.2 Convolutional Neural Network Architecture

The second model is a one-dimensional CNN that learns local temporal features from the predistortion input. It comprises:

- **Conv1D:** 6 filters of length 6, ReLU activation, input shape (20, 1).
- **AveragePooling1D:** pool size = 2.
- **Flatten:** converts pooled feature maps into a vector.
- **Dense:** 2 outputs, linear activation.

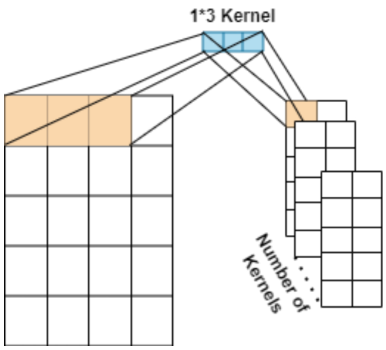

Figure 1. Convolution operation.

In our one-dimensional convolutional network, the core operation in the first layer is a discrete convolution between the input sequence and a set of learnable kernels (Fig. 1). Denoting the input at time step n as $x_n \in \mathbb{R}$ (reshaped here to $\mathbf{x} \in \mathbb{R}^{20\times1}$) and a single kernel of length $K = 6$ as

$$\mathbf{w} = (w_0, \ldots, w_5),$$

the convolution output at position i for filter f is

$$(\mathbf{x} * \mathbf{w}^{(f)})_i \;=\; \sum_{k=0}^{5} w_k^{(f)}\, x_{i+k} \;+\; b^{(f)},$$

where $b^{(f)}$ is the bias for filter f. With 6 such filters, this produces 6 feature-map sequences of length $20 - 6 + 1 = 15$. The ReLU activation,

$$z_i^{(f)} = \max(0,\ (\mathbf{x} * \mathbf{w}^{(f)})_i),$$

introduces nonlinearity by zeroing out negative responses.

2.3 Recurrent Neural Network (RNN) Architecture

The third model employs a SimpleRNN layer to capture sequential memory effects in the predistortion input, followed by two fully-connected layers to produce the final predistortion corrections. It consists of:

- **SimpleRNN:** 30 hidden units, tanh activation, input shape (20, 1).
- **Dense:** 30 units, ReLU activation.
- **Dense output:** 2 units, linear activation.

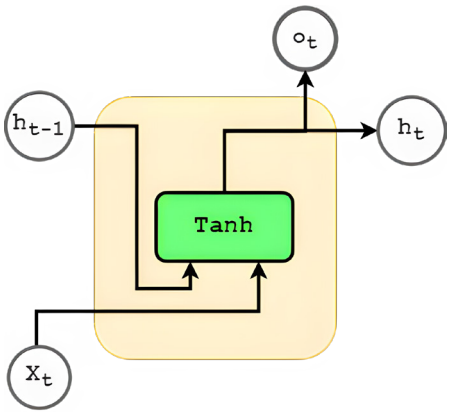

Figure 2. Structure of a Simple RNN cell.

As seen in Fig. 2, the RNN cell modifies its hidden state every time step by

$$h_t = \tanh(W_x\, x_t + W_h\, h_{t-1} + b),$$

where $W_h \in \mathbb{R}^{30\times30}$ and $W_x \in \mathbb{R}^{30\times1}$ are the hidden-to-hidden and input-to-hidden weight matrices, respectively, and $b \in \mathbb{R}^{30}$ is a bias vector. The network may simulate the memory effects of the power amplifier over the 20-sample window by sharing these parameters across time steps. The two predistortion output values are then produced by passing the final hidden state, h_{20}, through the thick layers.

2.4 Gated Recurrent Unit (GRU) Architecture

The fourth model uses two fully-connected layers to generate the final predistortion outputs after a Gated Recurrent Unit (GRU) layer conveys both short-term and long-term dependencies in the predistortion input. (Fig. 3). It comprises:

- **GRU:** 32 units, tanh activation for the candidate state, input shape (20, 1).
- **Dense:** 30 units, ReLU activation.
- **Dense output:** 2 units, linear activation.

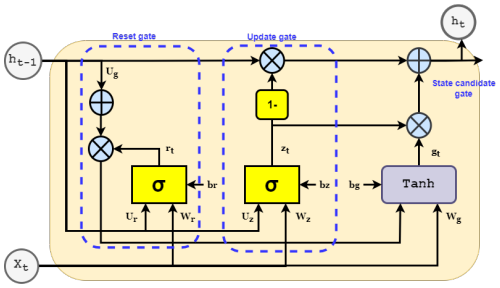

Figure 3. Structure of a GRU cell.

At each time step t, given input $x_t \in \mathbb{R}$ and earlier hidden state $h_{t-1} \in \mathbb{R}^{32}$, the GRU cell computes:

$$
\begin{aligned}
r_t &= \sigma(W_r x_t + U_r h_{t-1} + b_r),\\
z_t &= \sigma(W_z x_t + U_z h_{t-1} + b_z),\\
g_t &= \tanh(W_g x_t + U_g(r_t \odot h_{t-1}) + b_g),\\
h_t &= (1 - z_t) \odot h_{t-1} \;+\; z_t \odot g_t,
\end{aligned}
$$

where:

- r_t is The reset gate that decides how much of the earlier state should be forgotten.
- z_t is the update gate that balances the candidate state and the old state.
- g_t is the candidate hidden state.
- b_* are bias vectors and W_*, U_* are weight matrices.
- $\odot$ denotes element-wise multiplication,σ logistic sigmoid.

This gating technique effectively models the memory-dependent nonlinearities of the power amplifier by enabling the network to adaptively retain or discard information over the 20-sample window.

2.5 Long Short-Term Memory (LSTM) Architecture

The fifth model employs an LSTM layer to capture both short-term and long-term dependencies in the predistortion input, followed by two fully-connected layers to produce the final predistortion outputs (Fig. 4). It comprises:

- **LSTM:** 32 units, tanh activation for the cell update, default sigmoid gates, input shape (20, 1).
- **Dense:** 30 units, ReLU activation.
- **Dense output:** 2 units, linear activation.

At each time step t, given input $x_t \in \mathbb{R}$, previous hidden state $h_{t-1} \in \mathbb{R}^{32}$, and previous cell state $c_{t-1} \in \mathbb{R}^{32}$, the LSTM cell computes:

$$
\begin{aligned}
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad \text{(forget gate)},\\
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad \text{(input gate)},\\
\tilde{c}_t &= \tanh(W_g x_t + U_g h_{t-1} + b_g) \quad \text{(candidate cell state)},\\
c_t &= f_t \odot c_{t-1} \;+\; i_t \odot \tilde{c}_t \quad \text{(updated cell state)},\\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad \text{(output gate)},\\
h_t &= o_t \odot \tanh(c_t) \quad \text{(hidden state output)},
\end{aligned}
$$

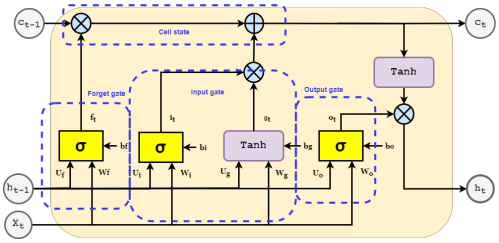

Figure 4. Structure of an LSTM cell.

where W_*, U_* are learnable weight matrices,σ is the sigmoid function,b_* are bias vectors, and $\odot$ indicates element-wise multiplication. This gating structure enables the LSTM to selectively preserve or reject information during the 20-sample window, effectively modeling the power amplifier's memory-dependent nonlinearities.

3 SIMULATION RESULTS

The overall evaluation setup is shown in Fig. 5.The CNN/GRU-DPD block receives an OFDM test signal with 16-QAM on each subcarrier, passes it through the power amplifier (PA) model, and then evaluates it using the adjacent channel power ratio (ACPR) and normalised mean squared error (NMSE).

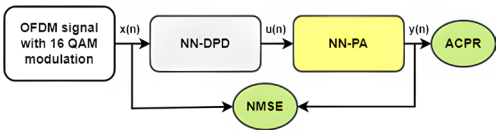

Figure 5. The DPD evaluation system's block diagram.

The dataset used for training and testing was provided by MathWorks [**?**]. It is made up of measured input and output waveforms from an NXP Airfast LD-59 MOS Doherty power amplifier, which is indicative of LTE/5G transmitters and works in the 3.6–3.8 GHz band with a gain of 29 dB. A 140 MHz, 5G-like OFDM waveform with 16-QAM modulation on each subcarrier serves as the test stimulus. The Adam optimiser was used to train all neural-network-based DPD models in order to minimise the mean-squared error between the ideal and predistorted PA outputs.

Table 1. PERFORMANCE COMPARISON OF DIFFERENT NN ARCHITECTURES

NN Architecture	L-ACPR (dBc)	R-ACPR (dBc)	NMSE (dB)	N° of Coefficients
No DPD	-29.39	-30.02	-21.05	–
DNN	-41.61	-42.38	-31.55	2552
CNN	-42.09	-42.17	-32.34	1394
RNN	-40.51	-40.69	-28.57	1952
GRU	-43.05	-42.86	-33.58	4412
LSTM	-42.60	-43.27	-33.63	5404

Table 1 summarizes the linearization performance achieved using different neural network (NN) architectures. The models are assessed using three key metrics: left and right Adjacent Channel Power Ratio (L-ACPR and R-ACPR), Normalized Mean Squared Error (NMSE), and the total number of learnable coefficients.

- **No DPD** represents the baseline case where no digital predistortion is applied. The ACPR values are relatively poor (around −30 dBc) and the NMSE is −21.05 dB, highlighting the strong nonlinear distortion introduced by the power amplifier.
- **FDNN** improves both ACPR and NMSE significantly compared to the no-DPD case. However, it requires a relatively large number of coefficients (2552), reflecting higher model complexity.
- **CNN** further improves NMSE (−32.34 dB) while using fewer parameters (1394 coefficients) than the DNN, demonstrating better efficiency in capturing the PA's memory effects through local convolutions.
- **RNN** provides moderate improvement in ACPR and NMSE compared to the no-DPD case, but its performance lags behind CNN, GRU, and LSTM. The RNN also has a relatively high number of coefficients (1952).
- **GRU** and **LSTM** architectures achieve the best NMSE and ACPR performance among all models, reaching NMSEs of −33.58 dB and −33.63 dB, respectively. This demonstrates the effectiveness of gated recurrent structures in learning both short and long term memory effects in the PA. However, this comes at the cost of increased model complexity, particularly for LSTM, which has the highest number of coefficients (5404).

Overall, the findings show a trade-off between performance and model complexity : although GRUs and LSTMs offer the best linearisation at the cost of greater parameter counts, CNNs offer a highly efficient solution with good linearisation performance and relatively low complexity.

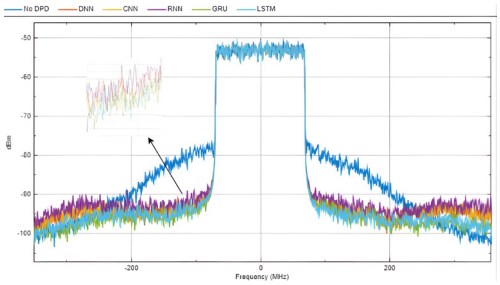

Figure 6. Output power spectral density (PSD) of the PA with and without DPD.

Fig. 6 displays the power amplifier's output power spectral density (PSD) under various predistortion techniques, compared to the case without DPD. The PSD is plotted in dBm versus frequency offset from the carrier.

The PA output shows notable spectrum regrowth without any predistortion (blue curve), with considerable adjacent channel leakage on both sides of the primary signal band. This spectral broadening may break spectral emission limits and interfere with nearby frequency channels.

The out-of-band emissions are greatly reduced by using neural network-based DPD. The spectrum regrowth is significantly reduced by all of the suggested NN-based predistorters,

including DNN, CNN, RNN, GRU, and LSTM. The best results are obtained by GRU and LSTM architectures, which push adjacent channel emissions below −90 dBm/Hz, suggesting strong linearisation. CNN also does well, offering robust suppression at a comparatively modest model complexity.

Overall, the figure confirms that all tested neural architectures enhance spectral efficiency, with GRU and LSTM offering the strongest adjacent channel protection.

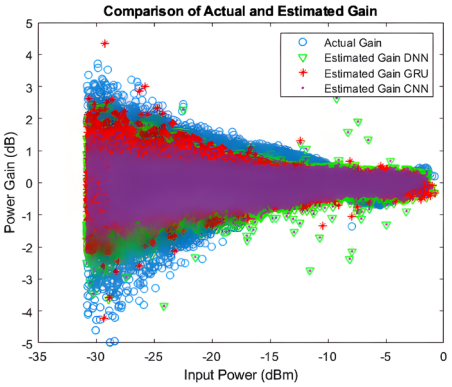

Figure 7. Comparison of the actual and estimated AM/AM characteristics for DNN-, CNN-, and GRU-based DPD models. For reasons of clarity, LSTM- and RNN-based results are not shown.

Fig. 7 displays the comparison of the real amplifier gain and the estimated gain after applying different neural network-based DPD models. For reasons of clarity, only the DNN-, CNN-, and GRU-based models are included in the figure; the LSTM- and RNN-based results are not shown. The power gain (in dB) is plotted versus the input power level (in dBm).

Without any predistortion, the blue circles show the power amplifier's true gain characteristic. At higher input power levels, where the amplifier becomes extremely nonlinear, significant gain compression is seen.

The estimated gain following the application of DNN-, GRU-, and CNN-based DPD is shown by the green triangles, purple dots, and red stars, respectively. When compared to the initial non-linear PA response, it is clear that all three neural network-based DPD models are successful in linearising the amplifier's behaviour and greatly lowering the gain compression.

4 Conclusion

In order to linearise a 5G-like OFDM signal, this study compared various neural network architectures for power amplifier digital predistortion. In comparison to the no-DPD condition, the results showed that all evaluated neural models significantly improve both NMSE and ACPR. CNN-based DPD outperformed the others in terms of performance and complexity, which makes it a desirable choice for real-world applications. At the expense of greater model size and training complexity, GRU and LSTM-based DPDs offered the best linearisation performance, with NMSE improvements surpassing 12 dB when compared to no predistortion. The capacity of recurrent architectures, particularly GRU and LSTM, to more accurately simulate memory effects and reduce out-of-band emissions was validated by power spectral density and AM/AM analysis. In order to better optimise the performance-complexity trade-off for next-generation wireless transmitters, further research could investigate hybrid topologies that combine convolutional and recurrent layers.

References

[1] A. Mohammadian and C. Tellambura, **RF impairments in wireless transceivers: Phase noise, CFO, and IQ imbalance—A survey,** IEEE Access, vol. 9, pp. 111718–111791, 2021.https://doi.org/10.1109/access.2021.3101845

[2] D. R. Morgan, Z. Ma, J. Kim, M. G. Zierdt, and J. Pastalan, **A generalized memory polynomial model for digital predistortion of RF power amplifiers,** IEEE Transactions on Signal Processing, vol. 54, no. 10, pp. 3852–3860, Oct. 2006.https://doi.org/10.1109/tsp.2006.879264

[3] S. Dikmese, L. Anttila, P. P. Campo, M. Valkama, and M. Renfors, **Behavioral modeling of power amplifiers with modern machine learning techniques,** in Proc. IEEE MTT-S Int. Microwave Conf. on Hardware and Systems for 5G and Beyond (IMC-5G), pp. 1–3, 2019. https://doi.org/10.1109/imc-5g47857.2019.9160381

[4] D. Wang, M. Aziz, M. Helaoui, and F. M. Ghannouchi, **Augmented real-valued time-delay neural network for compensation of distortions and impairments in wireless transmitters,** IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 1, pp. 242–254, Jan. 2018.https://doi.org/10.1109/tnnls.2018.2838039

[5] X. Hu, Z. Liu, X. Yu, Y. Zhao, W. Chen, B. Hu, X. Du, X. Li, M. Helaoui, W. Wang, et al., **Convolutional neural network for behavioral modeling and predistortion of wideband power amplifiers,** IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 8, pp. 3923–3937, Aug. 2021.https://doi.org/10.1109/tnnls.2021.3054867

[6] A. Li, H. Wu, Y. Wu, Q. Chen, L. C. N. de Vreede, and C. Gao, **DPDNeuralEngine: A 22-nm 6.6-TOPS/W/mm2 recurrent neural network accelerator for wideband power amplifier digital pre-distortion,** arXiv preprint, arXiv:2410.11766, 2024. https://doi.org/10.1109/iscas56072.2025.11043563

[7] A. Ambagahawela Rathnayake Mudiyanselage, **Augmented-LSTM and 1DCNN-LSTM based DPD models for linearization of wideband power amplifiers,** Master's thesis, A. Ambagahawela Rathnayake Mudiyanselage, 2023.https://doi.org/10.1109/pimrc56721.2023.10294019

[8] H. Imtiaz, Z. Zheng, R. H. Nejad, M. Zeng, and L. A. Rusch, Gated recurrent neural networks based pre-distortion for digital-to-analog converter, in Proc. IEEE Photonics Conf. (IPC), pp. 1–2, 2022.https://doi.org/10.1109/ipc53466.2022.9975578

[9] MathWorks, "Power Amplifier Characterization," https://www.mathworks.com/help/comm/ug/power-amplifiercharacterization.html, accessed Apr. 2025.