Artificial Neural Networks for GaN HEMT Model Extraction in D-band Using Sparse Data

Andrea Arias-Purdue, Eythan Lam, Jonathan Tao, Everett O'Malley, James F. Buckwalter
University of California, Santa Barbara, USA
andrea_arias@ucsb.edu

***Abstract*— We describe the application of Artificial Neural Networks (ANNs) for Gallium Nitride (GaN) High-Electron Mobility Transistor (HEMT) model parameter extraction to improve the model accuracy between 110 and 170 GHz. Fully-connected ANNs trained by backpropagation relate the physics-based ASM-HEMT model parameters to RF transistor measurements. The effects of ANN activation function, number of layers, number of nodes and number of training set data points on training accuracy are studied. For the 12 model parameters that dominate the 40-nm GaN HEMT RF characterization, we obtained a combined root-mean-squared (RMS) error of 2.5% between the ANN prediction and the training set, which is acceptable for most design tasks.**

***Keywords*— GaN, HEMT, ASM-HEMT, machine learning, artificial neural network, backpropagation, millimeter-wave.**

I. Introduction

GaN HEMTs are amongst the most promising devices to substantially increase power density in millimeter-wave bands for communications and radar [1]. Scaled GaN HEMTs exhibit high transistor figures of merit such as transit frequency (f_T) and maximum frequency of oscillation (f_{MAX}) while simultaneously achieving excellent breakdown voltage. In [2], the 40-nm T-gate process offers f_T/f_{MAX} = 200/400 GHz with a breakdown exceeding 40 V and has been shown for a 140-GHz, 23-dBm power amplifier (PA) [3]. Due to the limited available gain (below 6 dB) at D-band (110-170 GHz), a careful performance trade-off study must be undertaken to achieve the optimal loadline for power and/or power added efficiency (PAE). The design of a D-band PA requires precise tuning with the transistor internal parasitics to achieve desired performance and typically requires several iterations in III-V processes as illustrated in the conventional approach in Fig. 1.

Most III-V device model development is typically constrained to frequency bands under 60 GHz. Current control (I-V) and small-signal (S-parameters) inform model parameters in physics-based models while large-signal loadpull (LP) is often used for model validation. Hence, the device physics is fully characterized by the small-signal and DC data alone. Standard model extraction techniques (either physics-based or phenomenological) are linear in nature: the model parameters are extracted either through DC or AC tests, however most parameters are affected by both conditions.

Moreover, these models oftentimes become inaccurate at higher frequency bands where non-linear (power sweep) device data may not be available. In D-band, a comparison of simulation and measurement of the 40-nm GaN HEMT is shown in Fig. 2 and indicates model accuracy of around 8-15%, which is a significant uncertainty to design circuits that might be tuned over a 10% bandwidth.

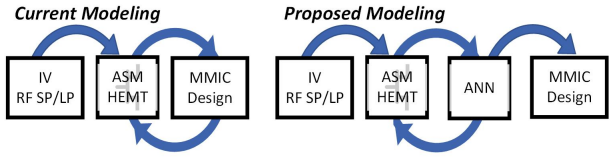

Fig. 1. Proposed ANN model modifies foundry models to fit measurement characteristics at higher frequency bands.

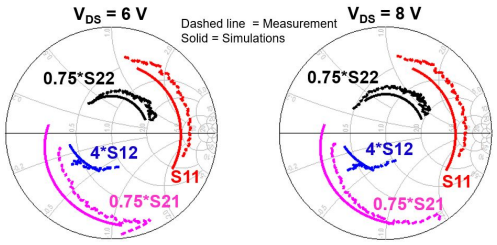

Fig. 2. Measured transistor S-parameters (dashed line) and simulations (solid line) from 110-170 GHz of a 40-nm GaN device (gate width W_g= 4x37.5 μm) at 0.2 A/mm. The accuracy of the model is around 8-15%.

This work proposes Artificial Neural Networks (ANNs) as part of a design methodology for rapid and accurate physics-based device model parameter extraction as illustrated in Fig. 1. THe ASM-HEMT model is emerging as a standard physics-based model, compatible with most circuit simulators [4]. Our approach leverages the ANN to map the non-linear interrelations between model parameters with DC and RF performance. Prior work has successfully used deep learning for DC-parameter model extraction, training to a total of 6 parameters [5].

The proposed methodology significantly extends the limitation of the prior work to develop an ANN model from sparse data in the form of basic DC and S-parameter characterization to complete a set of ASM-HEMT parameters. We identify 12 model parameters that most affect the model and demonstrate an RMS training and validation error of 2.3-2.7%. In section II, we review the appropriate choice of machine learning approaches for device modeling. We demonstrate the connection of the ANN to the ASM-HEMT model and the selection of hidden layers for an optimally-trained ANN. Section III describes the results of the training and improvements in the S-parameters measurement agreement between 110 and 170 GHz.

2023 IEEE Radio Frequency Integrated Circuits Symposium (RFIC) | 979-8-3503-2122-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/RFIC54547.2023.10186179

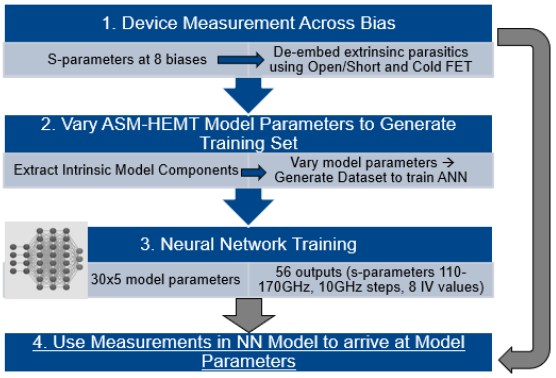

Fig. 3. Training methodology for ANN of the ASM HEMT model.

II. ANN-Based Modeling Methodology

Neural networks have been considered for transistor modeling since the early days of computer-based semiconductor device modeling [6] including small-signal [7]-[8], and large-signal modeling [9]-[10]. Various neural network architectures have been proposed for transistor modeling, including fully-connected convolutional networks (CNNs) [11], and recurrent networks (RNN) [12]. Here, we use the ASM-HEMT model, a potential-based physical model that has been shown to offer good agreement for GaN HEMTs [13] and that can be parameterized through the ANN training methodology summarized in Fig. 3.

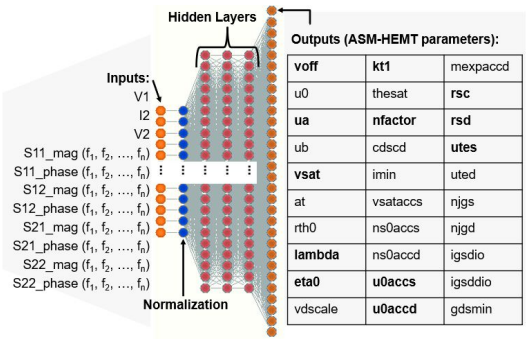

Fig. 4. ANN model for ASM-HEMT parameter extraction.

First, representative S-parameters measurements from 110-170 GHz are made to arrive at the intrinsic model parameters. These intrinsic parameters are used in the ASM-HEMT model and a total of 30 model parameters are varied (5 variations per parameter), with each simulation generating a set of outputs (8 IV values and S-parameters at 7 frequency steps from 110 - 170 GHz). To train the ANN, a dataset containing the model parameter inputs and corresponding IV and S-parameters at each of the 7 frequencies is generated while the ANN outputs are the 30 model parameters. The training datasets contain a varying amount of entries in multiples of 12k points (110-170 GHz in 10 GHz steps). The ASM-HEMT model simulations were performed using Keysight ADS, varying each of the 30 model parameters identified in Fig. 4 arranged in 6 sub-groups per bias point (V_D, I_D). Each simulation generated up to 115k points, which was found optimal in that it kept the simulation time under 2 minutes.

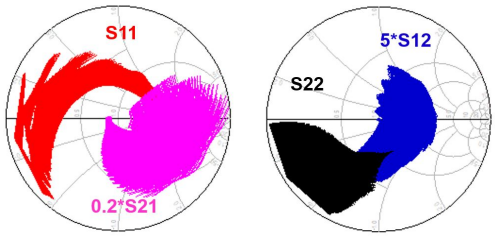

Fig. 5. Composite S-parameter plot of partial training set varying 6 model parameters at V_{DS} = 8 V and I_D = 20 mA.

The fully-connected neural network contains a normalization layer and a varying number of nodes and hidden layers (Fig. 4). Each node is indexed and updated independently from its layer, so that it can be easily de-activated without incurring additional computational losses and without having to modify the layer. The ANN inputs are trained using the backpropagation algorithm [14].

In the forward direction, the jth node output, Output_j, is calculated through an activation function using

$$\text{Output}_j = f_{\text{Activation}}(\text{Bias} + \Sigma(\text{Input}_i \times W_i)), \quad (1)$$

where W_i is the weight for the ith input. The weights are updated by applying a correction (ΔW_i) for output nodes:

$$\Delta W_i = \eta \times \Delta N_{\text{OUT}} \times f'_{\text{Activation}} \times \text{Input}_i + \alpha \times \Delta W_{i(n-1)} \quad (2)$$

where η is the training coefficient and

$$\Delta N_{\text{OUT}} = \text{Training Data - Output (Output Node)} \quad (3)$$

The derivative of the activation function ($f'_{\text{Activation}}$) is evaluated at the current training instance, Input_i is the ith input, α is the momentum coefficient, and $\Delta W_{i(n-1)}$ is the delta weight from the previous training instance.

For hidden nodes, output errors need to be backpropagated through the neural net. Eq. (2) still applies, however, ΔN_{OUT} is now

$$\Delta N_{\text{OUT}} = \Sigma(W_k \times \Delta N_{\text{OUT},k}) \text{ (Hidden Node)} \quad (4)$$

where k is the index of the kth output node connected to the hidden node.

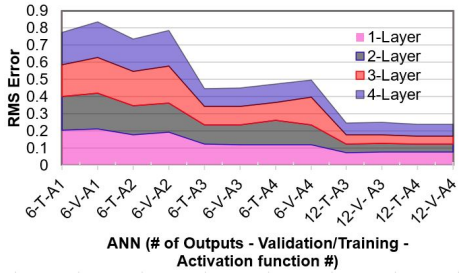

Fig. 6. Stack graph of RMS error performance of training (T) and validation sets (V) 1- through 4-layer ANNs, with activation functions 1 through 4, with 6 and 12 outputs. The activation functions used are: Tanh (A1), sigmoid + Gaussian (A2), sigmoid (A3), and ReLU + sigmoid (A4).

III. ANN Training Matrix and Results

The ANN described in the last section provides foundational insight into ANN structure, such as the number of hidden nodes and layers, activation functions and training coefficients. Our assumption is that semiconductor systems exhibit universal IV behavior signatures, manifested as saturation and pinch-off. The RF behavior is generally influenced by dynamic effects such as feedback capacitance and charge modulation that are also present in all devices to varying degrees. These effects can be globally and accurately captured by the appropriate activation functions, weights, bias and other ANN parameters.

Our goal is to optimize the ANN design in terms of its size (layer and node count), activation function, number of training set data points and training coefficient. The ANN performance parameters are the average RMS error relative to the training set and the training time. Commonly used activation functions were included in this study, including sigmoid, tanh, ReLU, Gaussian, arc(cos) and Shockley equation. Each activation function bias and leakage is randomized prior to training. The backpropagation training coefficient (η) was also varied from 0.2 to 0.0001 (where the baseline value is 0.002). The ANN design matrices are constructed by, first, completing a coarse version, followed by a finer set of ANNs with varying number of nodes, training set size and backpropagation coefficient. In addition, we also repeated some of the first matrix variations (e.g. number of layers, number of outputs and activation function) to ensure that the chosen design space was optimal.

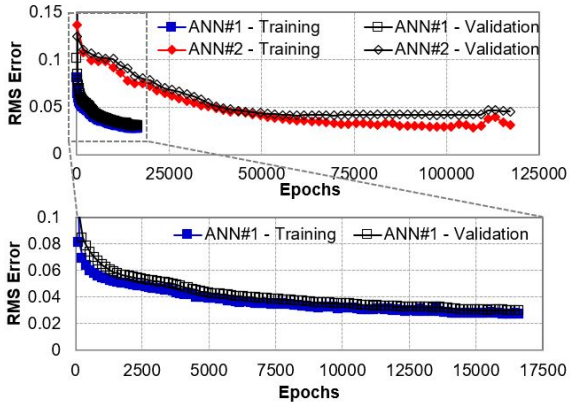

Fig. 7. RMS training error of downselected ANNs. ANN#1 trains significantly faster than ANN#2, while reaching the same overall RMS error of 2.3-2.7%.

Fig. 6 summarizes the results of the first ANN design matrix, showing a stack graph of each ANN version simulated. Each ANN has 16 nodes per layer. The same training data set comprised of 115k entries was used to understand the effect of the number of layers (fixed number of nodes per layer), the activation function and the number of outputs being learned relative to the number of inputs. Fig. 6 includes the training error and the validation set error for each ANN. The validation set is generated by selecting 10% of the training set data at random and removing it from the training/backpropagation parts of the algorithm. The simulations indicate that 2 to 3 layers appear to perform better as compared to 1- or 4-layer ANNs. In all cases, the RMS error of the validation set is very similar to that of the training set. We are able to calculate the ANN overall RMS error since the ANN numerical outputs are all normalized to the same limits. From Fig. 6, the choice of activation function has a noticeable effect on the RMS error, with functions A3 and A4 performing the best.

Next, we varied the number of nodes followed by the number of training set data points, from 115k to 128k and 140k. Varying the number of nodes per layer from 12 to 24 in steps of 4 revealed that RMS performance was maintained for the 12 - 20 nodes case while it deteriorated for the 24 node ANN for activation functions A2 and A3 (from 5% to 7%). Increasing the number of training data points from 115k to 140k did not improve RMS error, with the best performance being about 5% RMS error for both the training and the validation sets (same as baseline ANN obtained with the first design matrix). Likewise, varying the backpropagation coefficient η did not signifcantly change the error, although we note that if the parameter η is too large (>0.1) the ANN weights overflow to very large figures and the net cannot recover.

A 5% overall error is generally not sufficient to generate an accurate ANN model, hence we revised the ANN input structure next. The new structure key difference is in the frequency dependence of the data; instead of frequency being a free parameter with 7 entries, the ANN inputs were specified for each frequency trained, so the number of input nodes was increased from 12 (current, voltage, frequency and 8 S-parameters) to 59 (8 S-parameters per frequency point). The modified data array improved the RMS error by two-fold, from 5% (Fig. 6, best case) to 2.3-2.7% (Fig. 7). Further improvement can likely be effected by increasing the dataset bandwidth, although we will show that we can obtain robust model predictions based on our sparse model.

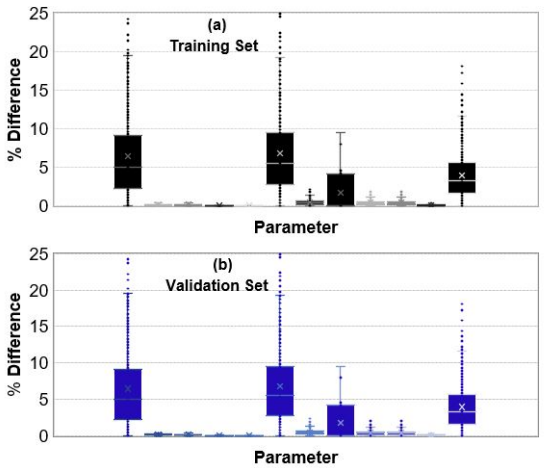

Fig. 8. Box plot graph of percent error of each of the 12 parameters tracked for the training (a) and validation sets (b).

A similar study varying activation function, layer and node count, training set size and backpropagation coefficient was performed on our revised ANN design (59 inputs) with the

resulting RMS averages mostly around 3%, confirming our findings during our first set of ANN simulations.

While non-optimal activation functions can increase RMS error, it appears that the appropriate ANN design space allows for a relaxed array of functions. We note that when attempting to achieve errors below 0.5%, activation function choice optimization likely becomes important. In our case, where the target is <2.5% RMS, ultimately, the RMS error lower bound appears to be limited by dataset structure. ANN design parameters such as activation function, backpropagation rate and other factors can improve RMS error but they mostly affect training time, an important efficiency metric.

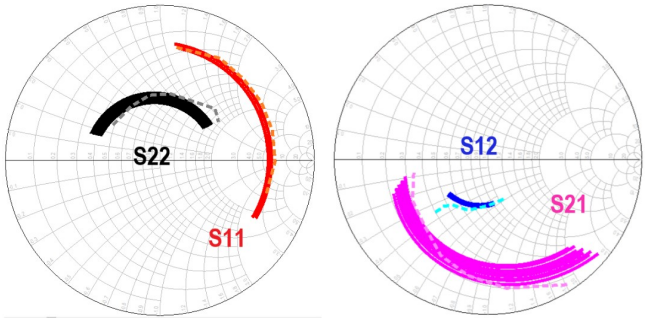

Fig. 9. Composite graph showing the ANN-generated model (solid) alongside measured S-parameters (dashed line) from 110-170 GHz at 0.2 A/mm (30 mA) V_D= 8 V bias.

The lowest error ANNs are demonstrated in Fig. 7. We highlight that an optimal activation function choice can significantly improve training time (under 2 minutes for ANN#1). ANN#1 uses a combination of ReLU and sigmoid function nodes, while ANN#2 uses sigmoid only. For ANN#1, the individual parameter RMS error averages are <1% for 6 parameters, between 1 and 5% for 2 parameters, between 5 and 8% for 2 parameters and 8-15% for the last 2 parameters (Fig. 8). Using the ANN-predicted model parameters and associated error, we simulated the device S-parameters and compared it against our D-band measurements. A composite plot is shown in Fig. 9 demonstrating reasonably good agreement compared to the Foundry model in Fig. 2.

IV. Conclusion

Through optimization of dataset structure, activation function, number of layers and nodes and backpropagation coefficient, we demonstrated an ANN model that successfully predicts the ASM-HEMT model parameters in the 110-170 GHz range at an overall RMS error of 2.3-2.7%. The ANN model training time was only 2 minutes, which compares very favorably to hours/days typically spent using standard fitting methods that often include manual model parameter optimization. The results highlight the importance of dataset structure and identifies various ANN design tradeoffs such as activation function and backpropagation rate as secondary contributors that can improve RMS error but mostly affect training time for the ANN design space analyzed in this work.

Acknowledgment

This work was supported by Defense Advanced Research Projects Agency (DARPA) through a DSSP under the SRC ComSenTer Program. The authors gratefully acknowledge DARPA program manager Dr. Tom Kazior's guidance and support through the grant that funded this effort. The authors thank GradientN for making their ANN software NNetDesigner© available for this work. The authors also thank HRL technical staff for valuable discussions and guidance and Dr. Tim Hancock from DARPA for facilitating access to the GaN foundry.

References

[1] Z. Popovic, "Amping up the pa for 5g: Efficient gan power amplifiers with dynamic supplies," *IEEE Microwave Magazine*, vol. 18, no. 3, pp. 137–149, 2017.

[2] K. Shinohara, A. Corrion, D. Regan, I. Milosavljevic, D. Brown, S. Burnham, P. J. Willadsen, C. Butler, A. Schmitz, D. Wheeler, A. Fung, and M. Micovic, "220ghz f_t and 400ghz f_{max} in 40-nm gan dh-hemts with re-grown ohmic," in *2010 International Electron Devices Meeting*, 2010, pp. 30.1.1–30.1.4.

[3] E. Lam, A. Arias-Purdue, E. O'Malley, and J. F. Buckwalter, "A 23.5-dbm, 7.9%-pae pseudo-differential power amplifier at 136 ghz in 40-nm gan," in *2022 17th European Microwave Integrated Circuits Conference (EuMIC)*, 2022, pp. 119–122.

[4] S. Ghosh, S. A. Ahsan, A. Dasgupta, S. Khandelwal, and Y. S. Chauhan, "Gan hemt modeling for power and rf applications using asm-hemt," in *2016 3rd International Conference on Emerging Electronics (ICEE)*, 2016, pp. 1–4.

[5] F. Chavez, D. T. Davis, N. C. Miller, and S. Khandelwal, "Deep learning-based asm-hemt i-v parameter extraction," *IEEE Electron Device Letters*, vol. 43, no. 10, pp. 1633–1636, 2022.

[6] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, 1982, 1982.

[7] H. Habal, D. Tsonev, and M. Schweikardt, "Compact models for initial mosfet sizing based on higher-order artificial neural networks," in *2020 ACM/IEEE 2nd Workshop on Machine Learning for CAD (MLCAD)*, 2020, pp. 111–116.

[8] P. Watson, M. Weatherspoon, L. Dunleavy, and G. Creech, "Accurate and efficient small-signal modeling of active devices using artificial neural networks," in *GaAs IC Symposium. IEEE Gallium Arsenide Integrated Circuit Symposium. 20th Annual. Technical Digest 1998 (Cat. No.98CH36260)*, 1998, pp. 95–98.

[9] A. Jarndal, S. Husain, M. Hashmi, and F. M. Ghannouchi, "Large-signal modeling of gan hemts using hybrid ga-ann, pso-svr, and gpr-based approaches," *IEEE Journal of the Electron Devices Society*, vol. 9, pp. 195–208, 2021.

[10] J. Xu and D. E. Root, "Artificial neural networks for compound semiconductor device modeling and characterization," in *2017 IEEE Compound Semiconductor Integrated Circuit Symposium (CSICS)*, 2017, pp. 1–4.

[11] S.-C. Han, J. Choi, and S.-M. Hong, "Acceleration of semiconductor device simulation with approximate solutions predicted by trained neural networks," *IEEE Transactions on Electron Devices*, vol. 68, no. 11, pp. 5483–5489, 2021.

[12] B. S. Paskaleva, V. Shitole, and E. Rosenbaum, "Data-driven compact modeling of bipolar junction transistors with recurrent neural networks." 8 2021. [Online]. Available: https://www.osti.gov/biblio/1884897

[13] A. Dasgupta, S. Ghosh, Y. S. Chauhan, and S. Khandelwal, "Asm-hemt: Compact model for gan hemts," in *2015 IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC)*, 2015, pp. 495–498.

[14] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.