

# Artificial neural network design for compact modeling of generic transistors

Lining Zhang<sup>1</sup> · Mansun Chan<sup>1</sup>

Published online: 9 April 2017  
© Springer Science+Business Media New York 2017

**Abstract** A methodology to develop artificial neural network (ANN) models to quickly incorporate the characteristics of emerging devices for circuit simulation is described in this work. To improve the model accuracy, a current and voltage data preprocessing scheme is proposed to derive a minimum dataset to train the ANN model with sufficient accuracy. To select a proper network size, four guidelines are developed from the principles of two-layer network. With that, a reference ANN size is proposed as a generic three-terminal transistor model. The ANN model formulated using the proposed approach has been verified by physical device data. Both the device and circuit-level tests show that the ANN model can reproduce and predict various device and circuits with high accuracy.

**Keywords** Compact model · Emerging device · Device modeling · Artificial neural network (ANN)

## 1 Introduction

Toward the end of the Moore's law and device scaling limit, many emerging devices are under extensive investigation for their potential to extend the benefit from technology scaling. To evaluate the advantages of these devices in different applications, a compact model for each device is necessary

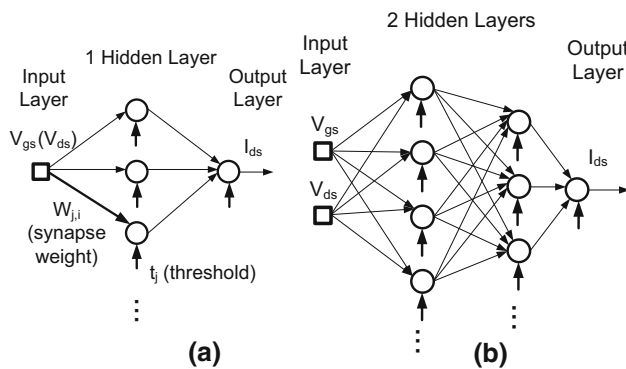
to allow computer simulation before the physical hardware is implemented. Due to the large number of devices being investigated, developing a physics-based model [1,2] for each device in the traditional approach is difficult to catch up with the rapid technology development.

In order to quickly incorporate newly generated device data into circuit simulations, data-oriented modeling methods have been used for technology evaluations without going into the detailed device physics. A table-based model is the most commonly used method in this category. For a given table constructed from device data, the accuracy is limited by the measurement noise and interpolation methods, especially in calculating the current derivatives. More accurate simulation can be achieved by increasing the data density, but it significantly increases the cost of data collection and simulation time. The limitations of table-based model have already been extensively discussed in [3]. Another data-oriented modeling approach is to construct an artificial neural networks (ANN) model [4,5], which has been used mainly for RF/microwave circuit optimizations [6,7]. An ANN model provides continuous and smooth approximation to the device data that eliminates the need for local interpolation in table-based models. However, an accurate ANN model over the entire operation regions of a transistor from subthreshold to strong inversion is still difficult to achieve. Furthermore, a standard and easy-to-use approach to select the required dataset and size of an ANN for the model training is not available. In order to use an ANN model in circuit simulations, convergence issues also need to be addressed. In this work, we have developed a guideline to design an ANN model including the method to select the minimal dataset for training. The model will be implemented into a circuit simulator to demonstrate its numerical stability and compatibility with existing simulation framework.

---

✉ Lining Zhang  
lnzhang@ieee.org; eelnzhang@ust.hk

<sup>1</sup> Department of ECE, Hong Kong University of Science and Technology, Kowloon, Hong Kong



**Fig. 1** Feed-forward networks are used to approximate the transistor physics and serve as a device model. As an example, with voltage inputs and current output, the ANNs of **a** one single hidden layer and **b** two hidden layers are potential methods for the current–voltage characteristics modeling

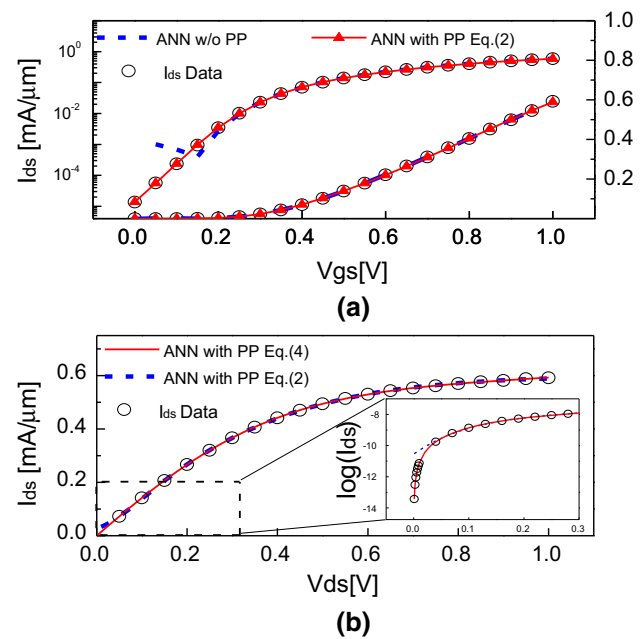
## 2 ANN compact modeling framework

The most common form of ANNs is the feed-forward network as shown in Fig. 1, which consists of an input layer, one or a few hidden layers of neurons and an output layer. According to a universal approximation theorem [4], such a network is capable of approximating a nonlinear function of multi-dimensional variables with its parameters consisting of the neuron synapse weights and thresholds. When the physics governing the transistor operation are unknown or complicated, an ANN can be used to approximate these physics equations. To construct a transistor model using an ANN, its terminal voltages (like  $V_{gs}$  and  $V_{ds}$ ) are used as **input** and the terminal currents (like  $I_{ds}$ ) and conductances (like the transconductance  $g_m$ ) are used as the **outputs**. Additional inputs like the device geometries and additional outputs like the device's terminal capacitance can also be incorporated into the network. After the size (the number of hidden layers and the number of neurons in each layer) is determined, the ANN is considered as a compact model and implemented into a circuit simulator.

Before using the ANN for circuit simulation, the ANN has to be trained to **minimize the mean square errors** between the output and the available device data by adjusting the network parameters. The process is similar to the model parameter extractions. A number of mature back-propagation (BP) algorithms [8] are available to train the ANN. With the availability of the ANN model and its parameters determined by the training process, the model is ready to be used for circuit simulation.

## 3 ANN model accuracy enhancement with data preprocessing and patterning

As the basis of compact modeling, an ANN with a single hidden layer is used to model the individual terminal's



**Fig. 2** Data preprocessing (PP) is necessary to improve the ANN model accuracy. **a** A logarithm current preprocessing improves the subthreshold accuracy. **b** A logarithm drain voltage preprocessing improves the linear region accuracy

control of the transistor's properties. Among the available ANN activation functions, **those of the sigmoid types with smooth derivatives are preferred due to the requirements on transistor's first- and higher-order derivatives from circuit simulations**. Two commonly used functions include the *tansig* and *logsig*. When it is applied to model the simple transfer characteristics ( $I_{ds}-V_{gs}$ ) of a MOSFET, the result is shown in the dash line in Fig. 2a. A network with two neurons and the *tansig* activation function in the hidden layer (denoted as 1-2-1 network to represent 1 input, 2 neurons in one hidden layer and 1 output) works well in the strong inversion region, but poorly in the deep subthreshold region. **Changing of the activation function or increasing the number of neurons of the hidden layer does not improve the model accuracy**. During the BP training, the network parameters are adjusted according to the output error. With the transistor's current data  $d(V_{gs})$ , the actual network output  $y(V_{gs})$ , the number of gate voltages in the dataset  $N$  and a learning rate  $\eta$ , the adjustments to the network parameters  $w$  in the training process are given by:

$$e(V_{gs}) = d(V_{gs}) - y(V_{gs})$$

$$\Delta w = -\frac{\eta}{N} \sum_{n=1}^N e[V_{gs}(n)] \cdot \frac{\partial e[V_{gs}(n)]}{\partial w} \quad (1)$$

**Adjustments from network errors in the subthreshold region are orders of magnitude smaller than those contributed from the above threshold region, thus have ignorable impacts on the training**. To solve this problem, a simple ANN model dataset preprocessing (PP):

$$[V_{gs}, d(V_{gs}) = \log(I_{ds})] \quad (2)$$

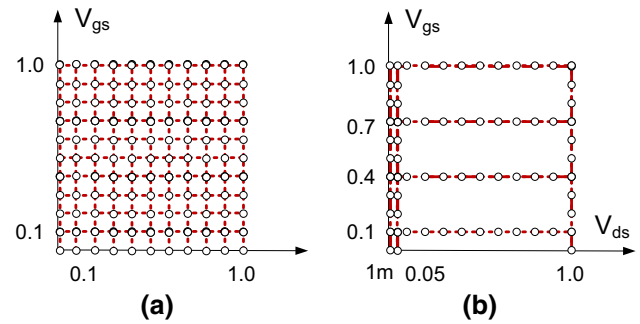
i.e., using the logarithm of transistors' current data as the network's target output is proposed. Due to the reduced target output scales, the same (1-2-1) network works perfectly in all the operation regions as shown by the solid line in Fig. 2a. However, the same preprocessing scheme fails the ANN modeling of the MOSFET output characteristics ( $I_{ds}-V_{ds}$ ). Figure 2b plots the results of another (1-2-1) network trained with  $\log(I_{ds})$  as the output target (the zero voltage and current data removed). Despite the overall accuracy, the ANN model gives a finite  $I_{ds}$  for  $V_{ds} = 0$ , violating the physical conservation law and will cause troubles in circuit simulations, e.g., the leakage power calculations. Forcing a finite current (even smaller than the SPICE current resolution) with  $V_{ds} = 0$  or adding one data point with  $V_{ds}$  close to zero does not work in the aspects that the trained ANN model gives an unreasonable channel resistance. More fine data points in the range of  $V_{ds}$  close to zero require an increase in the network size, resulting in over-fitting problems. Since most transistors show a linear  $I_{ds}-V_{ds}$  dependence with small drain voltages, the ANN target function becomes an exponential  $V_{ds}-\exp(\log(I_{ds}))$  that fails the BP training again. The drain voltage preprocessing with a logarithm is proposed to match the current preprocessing:

$$[\log(V_{ds}), d(V_{ds}) = \log(I_{ds})] \quad (3)$$

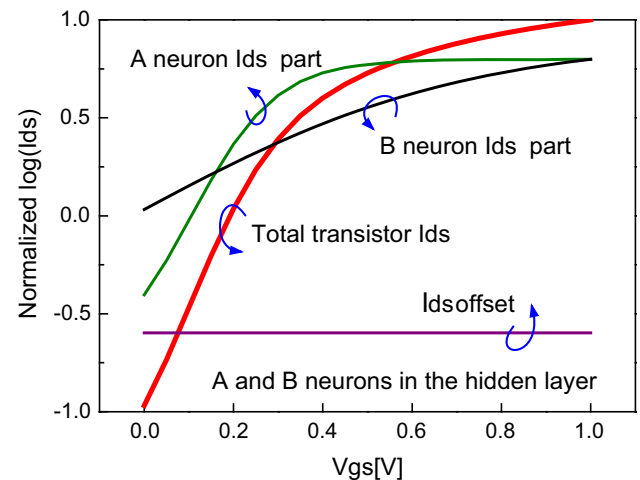
the logarithm of transistor's drain voltage is used as the network's input, with which the ANN target function resumes a linear one. A common practice in modeling which exchanges the source and drain when a negative  $V_{ds}$  is given is used here to avoid the logarithm of a negative number. For consideration of the ANN model accuracy in the linear region, one additional data point close to zero, e.g.,  $V_{ds} = 1$  mV, is included to the device dataset for training. The simple preprocessing in Eq. (3) is powerful to achieve higher ANN model accuracy that satisfies the SPICE simulation requirements. Figure 2b plots the ANN model (1-2-1) results (solid line) with the preprocessing scheme in Eq. (3). A SPICE zero (below the current resolution) is achieved in the small vicinity of zero  $V_{ds}$  as shown in the inset. For  $V_{ds} = 0$ , either an exact zero current is assigned or a smooth function [9] is implemented to guarantee an exact zero current. For a three-terminal transistor modeling, the final dataset preprocessing scheme is:

$$[V_{gs}, \log(V_{ds}), d(V_{gs}, V_{ds}) = \log(I_{ds})] \quad (4)$$

Depending on the data availability, different dataset patterns may be used in the ANN model training. Assuming a three-terminal transistor, a grid-like input dataset as shown in Fig. 3a is commonly used for a table-based model and



**Fig. 3** Two different datasets for the ANN-based compact device model. **a** A grid-like dataset with uniform steps in the terminal voltages. **b** A sparse dataset to represent the transistor characteristics which is similar to the one used for a physics-based model



**Fig. 4** A region-wise modeling concept is implemented in the ANN. With one neuron mainly working in the subthreshold region and another in the above threshold region, the transistor's transfer curve is reproduced

some ANN models. Since in the ANN training the dataset is usually divided into the train, test, and validation groups [8], smaller steps in the dataset are usually helpful for the model accuracy. As there is no guideline on the input step, a small step adopted results in a quite large dataset size of  $T = N^2$  where  $N$  is the number of data points along one terminal bias. As a result, it takes longer to obtain these data and to finish the BP training process. On the other hand, a sparse dataset similar to that in Fig. 3b is traditionally used for parameter extractions with a physics-based model [10]. The transistor's physics of its entire operation regions, including subthreshold, above threshold, linear, and saturation regions including the transitions in between, are fully represented by the sparse dataset. In principle, certain function extracted based on these sparse data can be regarded as an approximation to the transistor's physics.

The trained ANN model's characteristic for the transfer curve of Fig. 2a is plotted in Fig. 4. Among two neurons in the hidden layer, one neuron A approximates the car-

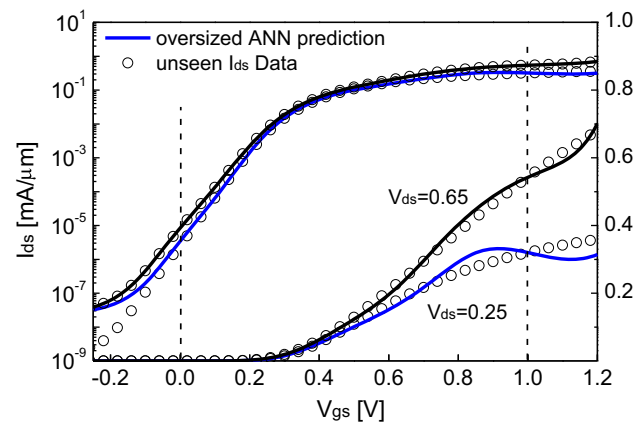
rier diffusion physics in MOSFET's subthreshold region, and the other neuron B approximates the carrier drift physics in the above threshold region. The output neuron linearly adds these two parts (with a constant offset) and provides the complete approximation to the MOSFET's transport physics. The sigmoid activation functions play similar roles as smoothing functions commonly used in compact models like the industry standard BSIM. Going to two-dimensional inputs of gate and drain voltages, the ANN's global approximation considers simultaneously the gate and drain dependence in transistors' carrier transports which are fully embedded in the sparse dataset in Fig. 3b. At the same time, the sparse dataset size is around  $T = 7N$ , practically smaller than the grid dataset. This transistor's knowledge-based dataset patterning is used in the following work.

#### 4 ANN model accuracy enhancement with a sizing technique

For a complete three-terminal MOSFET modeling, merging of the above two separate (1-2-1) networks for the transfer and output characteristics into a (2-4-1) network does not work with either the grid-like or the spares dataset. Indeed, training an ANN model of (2-X-1) fails even with an extremely large number of neurons in the hidden layer, which poses an intrinsic limitation of the single layer ANNs for compact modeling. Here an ANN with two hidden layers as another function approximation algorithm [11] is used for generic transistor compact modeling.

Without a general guideline for ANN sizing, an example ANN with two hidden layers (denoted as 2-8-4-1 to represent two inputs, 8 neurons in the first hidden layer, 4 in the second hidden layer, and 1 output) is applied to model one three-terminal transistor. The sparse dataset in Fig. 3b and the preprocessing scheme of Eq. (4) are used. While the standard Levenberg–Marquardt BP training is finished successfully, an application of the obtained model with two drain voltages that do not appear in the training dataset is shown in Fig. 5. Large errors in the transistor's linear regions are observed. The negative transconductance also possibly results in none convergence issues in SPICE simulations. These shortcomings are categorized as an over-fitting-induced generalization problem caused by too many neurons in the network. It is further confirmed with the model oscillation beyond the dataset range (outside of the dashed lines in Fig. 5) due to the superposition of outputs from more than necessary neurons. A sizing technique specifically for accurate transistor modeling is developed as follows.

The two-layer structure embeds the local and global features in the first and second hidden layers, in contrast to the monotonously global feature of ANNs with one hidden layer. Generally, MOSFETs have two distinct operation regions in

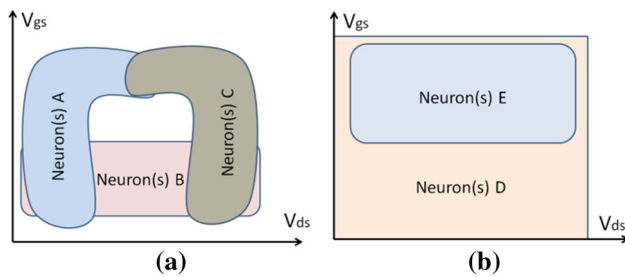


**Fig. 5** Without proper sizing, an ANN model with more than necessary neurons has large errors for inputs beyond the dataset and possible negative conductance within the operation voltage. Dashed lines show the boundary of trained gate voltages

the gate voltage dimension (subthreshold and superthreshold) and another two in the drain voltage dimension (before saturation and after saturation) that require separate mathematical descriptions. Correspondingly, in the ANN design, neurons in the first hidden layer are designed to mainly handle the gate and drain voltage inputs partition, while neurons in the second layer are designed to reproduce the regional characteristics and link them together. Figure 6 shows a possible task division for the neurons in each hidden layer to implement the local and global device characteristics. Similar to the input partition of the subthreshold and above threshold in Fig. 4, the neurons (or group) A–C divide the two-dimensional inputs, e.g., neurons A and B work in the subthreshold prior to drain saturation, while neuron(s) C will take over the drain saturation region. The regional characteristics are similar to the drift or diffusion properties in Fig. 4, and it is expected that at least four neurons are needed for roughly four operation regions. Neurons in the second hidden layer switch between the divided operation regions and provide the regional outputs. Figure 6b shows one example that the neuron(s) E only contributes to the above threshold region. At the same time, smooth transitions at the local region boundaries are promised by the neuron activation functions, forming the global features. More than two neurons are expected considering the interactions of the gate and drain voltages in transistor physics, e.g., the saturation drain voltage depends on the gate voltage. The similarity between the ANN fundamental principle (superpositions of continuous sigmoidal functions to form the complete functional space [12]) and the transistor physics (superpositions of drift/diffusion transports or distinguishable carrier statistics of ballistic transport) motivates the use of ANNs with two hidden layers for the transistor modeling.

A sensitivity analysis quantizes the above ANN designs. The neuron sensitivity is defined as the partial derivative





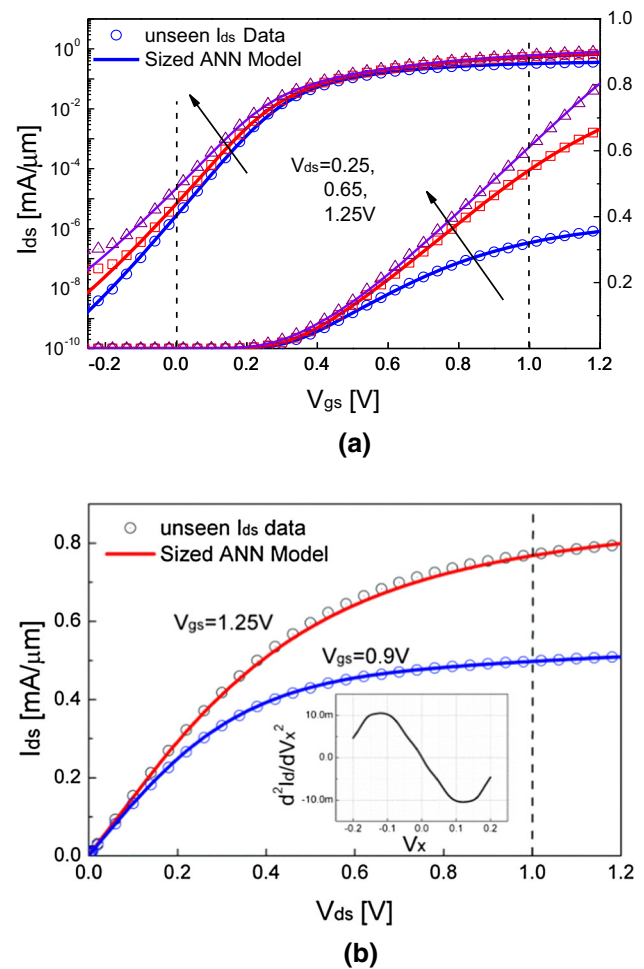
**Fig. 6** An ANN model with two hidden layers for a generic three-terminal transistor modeling. **a** The local regions partition is done in the ANN first hidden layer. **b** Neurons in the second hidden layer connects the local outputs smoothly to globally determine the transistor's current

of the network output over one specific neuron output, and derived based on the chain rule. With  $d$  as the output,  $C_{i,j}$  is the  $i$ th neuron output at the  $j$ th layer,  $g_1$ ,  $g_2$  and  $g_o$  are the first, second hidden layer, and the output layer activation functions,  $w$  is the network parameter (synapse weights) with the superscript showing the layer (hidden layer or the output layer) and the subscript showing the neuron's index, the neuron sensitivity is derived based on the chain rule:

$$\frac{\partial d}{\partial C_i^j} = g_o' \cdot \left[ w_p^o g_2' \cdot w_{pk}^2 \right], [i, j] = [p, o] \text{ or } [k, 2] \quad (5)$$

The network parameter matrix is assigned to implement the local and global control in the first and second hidden layers, respectively, over the output. The sensitivity is large if the input combination falls into the local region handled by the  $i$ th neuron, but small otherwise. The smooth transitions for the global feature are reflected in the choices of the  $g$  activation functions which are always infinitely differentiable sigmoidal functions. In the practical training, each specific neuron is randomly assigned the task as shown in Fig. 6 which does not affect the final results.

The above knowledge-based sizing technique suggests an ANN of (2-4-2-1) as the initial guess. Starting from that and increasing the number of neurons for accuracy improvement, it turns out that ANN with the size of (2-4-4-1) works for generic transistor modeling. Due to the wide data availability, FinFETs are used here to represent an emerging device. Figure 7 plots the generalizations of the ANN model trained for FinFETs (30 nm gate length). The activation functions in the first and second hidden layers are chosen as *tansig*, *logsig*, respectively. Considering that the *tansig* is just the *logsig* function biased and rescaled, or vice versa, they are equivalent from the mathematic modeling perspective. The short-channel effects, mobility degradations, and velocity saturations observed in the transistor are all reproduced. The model is accurate in reproducing the FinFET current–voltage characteristics, the conductance and transconductance, and higher-order derivatives. At the same time, the accuracy is also achieved with model extrapolations to voltage biases



**Fig. 7** A properly sized ANN model of (2-4-4-1) with 37 parameters exactly reproduces a short-channel FinFET characteristics. The model has good generalizations for **a** transfer curve and **b** output curves within the trained bias range, and can also extrapolate well 0.2 V beyond the operation voltage. The inset shows the Gummel symmetry tests

beyond the dataset range. The device data in Fig. 7 are collected on purpose for generalization validations of the reference model, in addition to the sparse dataset for model training in Fig. 4b. They are not necessary for general ANN training.

The ANN sizing is accompanied with a model verification step, similar to the common device modeling practice. With increasing the number of neurons, each ANN is trained and the obtained model is examined. The examinations include four benchmarks:

1. Mean square error (MSE) of the trained ANN model against the sparse dataset. MSE of the trained ANN model is defined similar to that of a physics-based model:

$$\text{MSE} = \sqrt{\frac{1}{T} \sum_{i=1}^T \left( \frac{y_i(V_{gs}, V_{ds}) - d_i(V_{gs}, V_{ds})}{I_{\text{threshold}}} \right)^2} \quad (6)$$

where  $I_{\text{threshold}}$  is a custom current (e.g., the maximum current in the dataset). For a noiseless dataset, a critical MSE of 0.2% is found for well-trained ANN models.

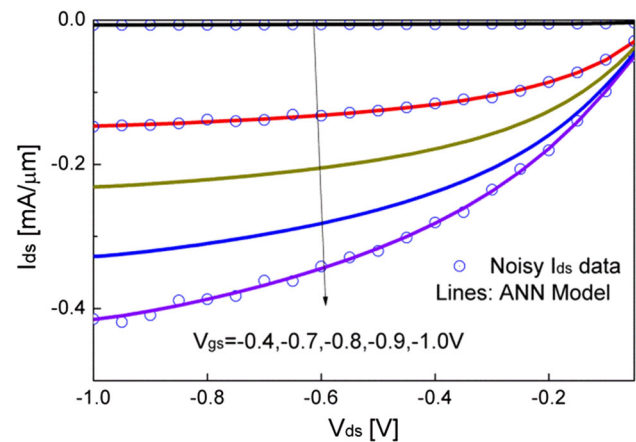
- Monotonicity and accuracy of the model first-order derivative against the sparse dataset. A transistor's conductance and transconductance is analytically derived assuming  $IN_i$  as the  $i$ th input:

$$\frac{\partial d}{\partial IN_i} = g'_o \cdot \left[ w_{jp}^o g'_2 \cdot \left( w_{pk}^2 g'_1 \cdot w_{ki}^1 \right) \right] \quad (7)$$

Full expressions of the device conductance are obtained by including the preprocessing functions and the input/output normalizations in the chain rule of Eq. (7). While a major transistor category shows positive conductance and transconductance over its operation regions, the ANN model first-order derivative is checked into a positive sign.

- Monotonicity of the model first-order derivative in interpolation and extrapolations. An ANN model with a least number of neurons gives monotonic first-order derivatives for inputs beyond the sparse dataset in Fig. 4b (all the interpolated inputs and certain extrapolated inputs, e.g., beyond 20% of the supply voltage).
- Sum of squares of the ANN model parameter  $w$ . The Bayesian regularization (BR)-based BP methods incorporate a philosophy that the sum of squares of the ANN model parameters is an indicator for the ANN model generalization ability. The smaller this sum is, the lesser chance for the over-fitting problem [8].

The obtained ANN of two hidden layers (2-4-4-1) is a reference for a generic three-terminal transistor model. This ANN is applied to model one p-type transistor with noisy characteristics considering that the measured device is always mixed with noise. Figure 8 shows that the ANN model provides a smooth fitting to the noisy data. No additional data preprocessing or smoothing is needed besides the one in Sect. 3. It is more important to apply the BR-based back-propagation method to avoid over-fitting to the noisy data. After training, the device conductance/transconductance is still available and smooth. Furthermore, the above ANN model should work for other emerging devices since it does not distinguish the transistor physics. As verified by other different transistors' data, the ANN model of (2-4-4-1) works for steep slope tunneling FETs [9] as one additional example. Slight adjustment to the ANN size may be subject to the emerging device behaviors with the above sizing technique.



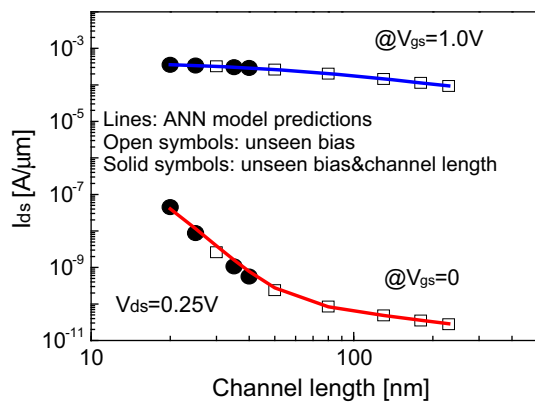
**Fig. 8** Reference ANN of (2-4-4-1) works when the training dataset in Fig. 3b is mixed with noise. A continuous and smooth model is obtained in which the derivatives are still available and not affected by the noise

## 5 ANN model in circuit simulations

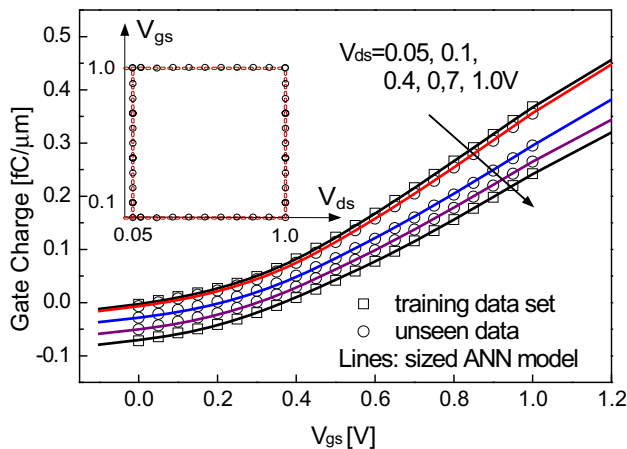
The above ANN model is extended for practical circuit simulations by considering the model feature enrichments and implementations into circuit simulators.

First, an ANN model is developed for multiple FinFETs with length scaling (from 30 to 230 nm). Device physics indicates that more complex mathematics are involved around the minimum gate length due to short-channel effects and so on, a smaller step in the gate length dimension is needed when constructing the training dataset. This knowledge-based data collection follows the same philosophy as in parameter extractions of a physics-based model [10]. In total, six lengths (30, 50, 80, 130, 180, 230 nm) are sampled and the sparse dataset in Fig. 3b for each transistor gate length is combined as the training data. Since gate length is the third model input besides two terminal voltages, the initial guess of the ANN size becomes (3-4-4-1). It turns out that with only one more extra neuron in each hidden layer, one ANN model of (3-5-5-1) reproduces the complex FinFET scaling behaviors on the short-channel effects, the carrier velocity saturations, etc. Figure 9 plots the generalization properties of such a generic ANN model in both the voltage and geometry dimension. Aided by the proposed four benchmarks, the obtained ANN model has superior generalization abilities in both the voltage dimension and gate length dimension, and has predictive abilities for further scaling. The overall MSE according to Eq. (1) reaches 1%, which is a challenging task for other modeling approaches.

Another generic ANN model is trained for a transistor's gate and drain terminal charge, as shown in Fig. 10. The dataset patterning of the total terminal charge (including both the intrinsic and extrinsic charge) is shown as the inset. Similar to the sparse dataset in Fig. 4b, a nongrid pattern is identified based on the knowledge about the terminal charge.



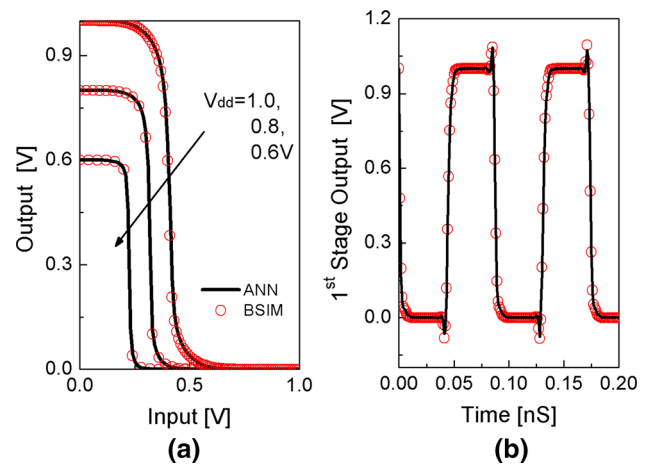
**Fig. 9** With the ANN sizing technique, a generic ANN model of (3-5-5-1) reproduces the transistor length scaling effect. The model generalizes well and has predictive ability, with just one more neuron in each hidden layer based on the reference model



**Fig. 10** An ANN model of (2-4-2) is developed to reproduce the gate charge of a short-channel transistor with the network sizing technique. High accuracy is achieved within and beyond the operation voltage. Inset shows the sparse dataset for training

The developed model pruning technique is practiced to obtain the number of neurons. An ANN model (2-4-2) is confirmed for modeling the gate and drain charge simultaneously. The model generalizes well. Terminal capacitances are obtained with the ANN derivatives similar to Eq. (7) that can be used together with the terminal charge model in circuit simulators.

With these features, a reasonable amount of parameters, and the powerful modern ANN training algorithms, a complete compact model for generic transistors covering the interested features is readily applicable in circuit simulations. The conductance and transconductance from a compact model required by the circuit simulators like SPICE is available once the ANN model is trained. They are regular expressions and higher-order differentiable. At the same time, a well-behaved compact model in a circuit simulator is able to handle bias voltages from negative infinite to positive infinite for the convergence of Newton iterations.



**Fig. 11** Circuit simulations with the ANN model in comparison with the BSIM. **a** DC simulation results of a CMOS inverter. **b** Transient simulation results of a 17-stage ring oscillator (RO). High accuracies are achieved

While this is a common issue for compact model developments [13] including physics-based models, the ANN model naturally satisfies this requirement due to the self-saturating sigmoidal activation functions. In other words, the limiting functions in a physics-based model are embedded in the ANN model itself. In case of the abnormal bias voltages when those derivatives are close to zero, the minimum conductance method is used for the simulation convergences.

The ANN model has been implemented into a simulator with Verilog-A and gone through simulation convergence and accuracy tests. With a sparse dataset in Fig. 3b generated by the BSIM-CMG model (ver.108.0.0) [14], an ANN model with the size of (2-4-4-1) is trained. The DC simulations of one CMOS inverter and the transient simulations of one 17-stage ring oscillator are shown in Fig. 11. Results of the ANN model agree well with those of the BSIM-CMG model. The designed ANN model works well in circuit simulations with high accuracy benchmarked with a physics-based model.

## 6 Conclusion

This work designs an artificial neural network for compact modeling of generic transistors. One scheme of data preprocessing is proposed and verified for the essential model accuracy enhancement. Together with the proposed sparse dataset patterning and the ANN sizing techniques, a standard and easy-to-use compact modeling approach is demonstrated. Its flexibility and accuracy for generic transistors in circuit simulations is verified after the ANN model implementations into a circuit simulator.

**Acknowledgements** This work was supported by the Hong Kong's University Grant Committee via the Area of Excellence project AoE-P04-08.

## References

1. Tsividis, Y.: Operation and Modeling of the MOS Transistor, 2nd edn. McGraw-Hill, New York (1999)
2. Khakifirooz, A., Nayfeh, O.M., Antoniadis, D.A.: A simple semiempirical short-channel MOSFET current–voltage model continuous across all regions of operation and employing only physical parameters. *IEEE Trans. Electron Devices* **56**(8), 1674–1680 (2009)
3. Root, D.E., Xu, J., Horn, J., Iwamoto, M.: The large-signal model: theoretical foundations, practical considerations, and recent trends. In: *Nonlinear Transistor Model Parameter Extraction Technique*, ch. 5. pp. 123–170. Cambridge University Press, Cambridge (2011)
4. Haykin, S.: *Neural Networks: A Comprehensive Foundation*, 2nd edn. Prentice-Hall, Englewood Cliffs (1999)
5. Root, D.E.: Future device modeling trends. *IEEE Microw. Mag.* **13**, 45–59 (2012)
6. Zhang, Q.J., Gupta, K.C.: *Neural Networks for RF and Microwave Design*. Artech House, Norwood (2000)
7. Xu, J., Yagoub, M.C.E., Ding, R., Zhang, Q.J.: Exact adjoint sensitivity analysis for neural based microwave modeling and design. *IEEE Trans. Microw. Theory Tech.* **51**(1), 226–237 (2003)
8. Hagan, M.T., Demuth, H.B., Beale, M.H., Jesus, O.D.: *Neural Network Design*, 2nd edn (2014)
9. Zhang, L., Chan, M.: SPICE modeling of double-gate tunnel-FETs including channel transports. *IEEE Trans. Electron Devices* **61**(2), 300–307 (2014)
10. Cheng, Y., Jeng, M.-C., Liu, Z., Huang, J., Chan, M., Chen, K., Ko, P.K., Hu, C.: A physical and scalable  $I$ - $V$  model in BSIM3v3 for analog/ digital circuit simulation. *IEEE Trans. Electron Devices* **44**(2), 277–287 (1997)
11. Barron, A.: Neural networks: a review from statistical perspective. *Statist. Sci.* **9**(1), 33–35 (1994)
12. Cybenko, G.: Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst.* **2**, 303–314 (1989)
13. McAndrew, C.C.: Practical modeling for circuit simulation. *IEEE J. Solid State Circuits* **33**(3), 439–448 (1998)
14. Khandelwal, S., Duarte, J.P., Venugopalan, S., Paydavosi, N., Lu, D.D., Lin, C.-H., Dunga, M., Yao, S., Morshed, T., Niknejad, A., Hu, C.: BSIM-CMG108.0.0 Technical Manual. (Online). [http://wwwdevice.eecs.berkeley.edu/bsim/?page=BSIMCMG\\_LR](http://wwwdevice.eecs.berkeley.edu/bsim/?page=BSIMCMG_LR) (2015). Accessed 01 Sept