Neural- Network-Based Charge Density Quantum Correction of Nanoscale MOSFETs

Li Zunchao¹, ‡, Jiang Yaolin², and Zhang Ruizhi¹

(¹ School of Electrical and Communication Engineering, Xi'an Jiaotong University, Xi'an 710049, China)
(² School of Science, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: For the treatment of the quantum effect of charge distribution in nanoscale MOSFETs, a quantum correction model using Levenberg-Marquardt backpropagation neural networks is presented that can predict the quantum density from the classical density. The training speed and accuracy of neural networks with different hidden layers and numbers of neurons are studied. We conclude that the high training speed and accuracy can be obtained using neural networks with two hidden layers, but the number of neurons in the hidden layers does not have a noticeable effect. For single and double-gate nanoscale MOSFETs, our model can easily predict the quantum charge density in the silicon layer and agrees closely with the Schrödinger-Poisson approach.

Key words: neural network; quantum correction; nanoscale MOSFET; charge density

1 Introduction

With advances in ULSI, MOSFETs are shrinking to the nanoscale regime, in which the dimensions are close to the De Broglie wavelength of the charge carriers. Quantum effects are evident, and the inversion layer charge carriers shift away from the SiO₂/Si interface — effects which must be considered in device modeling and simulation. The Schrödinger-Poisson equations with appropriate boundary conditions can be applied to study such quantum effects, but this is a time-consuming task in practice. In this paper, a back-propagation neural network (BP NN) is applied to construct a predictive model for the quantum correction of nanoscale MOSFETs that can predict the quantum charge density from the classical density. Though the standard gradient descent algorithm for BP NNs provides an easy learning method, it has three obvious drawbacks. First, it might converge to some local minimum. Second, initial weights and biases influence the learning speed. Third, it converges very slowly when the output is close to one. In this investigation, the output (the ratio of the quantum charge density to the classical density) is close to one when the point is far away from the SiO₂/Si interface. The Levenberg-Marquardt algorithm is used to avoid these drawbacks.

2 Quantum correction model

As illustrated in Fig. 1, the NN output layer has one neuron whose output denotes the ratio of quantum to classical charge density. The neurons in the input layer denote parameters such as oxide thickness, silicon layer thickness, gate voltage, and doping level and depth (distance from the SiO₂/Si interface), which determine the charge density ratio. There are some intermediate layers called hidden layers (first layer and second layer shown in Fig. 1).

The neurons in the input layer receive external inputs, and their weighted sums are transferred to the neurons in the first hidden layer. The input \( n^m_i \) of neuron \( i \) in hidden layer \( m \) is

\[
  n^m_i = w^m_i a^{m-1}_i + w^m_{i2} a^{m-1}_2 + \ldots + w^m_{i,m-1} a^{m-1}_{m-1} + b^m_i
\]

(1)

where \( a^{m-1}_i, a^{m-1}_{i2}, \ldots, a^{m-1}_{i,m-1} \) are the outputs of
the neurons in hidden layer \( m - 1 \), \( S^{m-1} \) is the number of neurons in hidden layer \( m - 1 \), \( w_{ij}^{m} \) is the weight between neuron \( j \) in hidden layer \( m - 1 \) and neuron \( i \) in hidden layer \( m \), and \( b_i^m \) is the bias of neuron \( i \) in hidden layer \( m \). The output \( a_i^m \) of neuron \( i \) in hidden layer \( m \) is
\[
a_i^m = f'(n_i^m) = 2/(1 + \exp(-2n_i^m)) - 1 \quad (2)
\]
where \( f' \) is the activation function.

The outputs \( a_1^m, a_2^m, \ldots, a_{n}^m \) of the neurons in hidden layer \( m \) are transferred to the neurons in hidden layer \( m + 1 \) and their weighted sums act as the inputs. The weighted sum of the outputs of neurons in the last hidden layer acts as the input to the neuron in the output layer. The activation function \( f^0 \) of the neuron in the output layer takes a linear form.

The network modeling capability is specified by the mean square error (MSE) of the output in the output layer as
\[
\text{MSE} = \frac{1}{Q} \sum_{q=1}^{Q} (t_q - a_q)^2 = \frac{1}{Q} \sum_{q=1}^{Q} e_q^2 \quad (3)
\]
where \( Q \) is the number of training vectors, and \( t_q \), \( a_q \), and \( e_q \) are the expected output, computed output, and the error for training vector \( q \), respectively.

In order to obtain the expected output for any external inputs, NNs need to be trained many times using several training vectors consisting of inputs and the corresponding outputs to determine the weights \( w_{ij}^m \) and biases \( b_i^m \) corresponding to the highest prediction accuracy.

In the training process, the weight and bias vector \( x \) is adjusted by [17]
\[
\Delta x_k = -(J^T(x_k)J(x_k) + \mu I)^{-1}J^T(x_k)e(x_k) \quad (4)
\]
where \( k \) is the iteration number, \( J \) is the Jacobian matrix of the error vector \( e \) to weight and bias vector \( x \), \( I \) is a unit matrix, and \( \mu \) is a scalar quantity used for controlling the search direction and step. \( e, x, \) and \( J \) are given in Eqs. (5) and (7), respectively.
\[
e^T = [e_1\ e_2\ \ldots\ e_q] \quad (5)
\]
\[
x^T = [x_1\ x_2\ \ldots\ x_q] \quad (6)
\]
Here \( R \) is the number of neurons in the input layer, \( M \) denotes the output layer, and \( b_i^m \) is the bias of the neuron in the output layer.
\[
J(x) = \begin{bmatrix}
\frac{\partial e_1}{\partial w_{1,1}} & \frac{\partial e_1}{\partial w_{1,2}} & \ldots & \frac{\partial e_1}{\partial w_{1,q}} & \frac{\partial e_1}{\partial b_1} & \ldots & \frac{\partial e_1}{\partial b_M} \\
\frac{\partial e_2}{\partial w_{1,1}} & \frac{\partial e_2}{\partial w_{1,2}} & \ldots & \frac{\partial e_2}{\partial w_{1,q}} & \frac{\partial e_2}{\partial b_1} & \ldots & \frac{\partial e_2}{\partial b_M} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\frac{\partial e_q}{\partial w_{1,1}} & \frac{\partial e_q}{\partial w_{1,2}} & \ldots & \frac{\partial e_q}{\partial w_{1,q}} & \frac{\partial e_q}{\partial b_1} & \ldots & \frac{\partial e_q}{\partial b_M}
\end{bmatrix}
\quad (7)
\]

The elements of \( J \) are computed by Eqs. (8) and (9).
\[
[J]_{h,1} = \frac{\partial e_h}{\partial x_1} = \frac{\partial e_h}{\partial w_{1,j}} = u_{i,h} m \times a_j^{-1} \quad \text{for weight } x_i \quad (8)
\]
\[
[J]_{h,1} = \frac{\partial e_h}{\partial x_1} = \frac{\partial e_h}{\partial b_1} = u_{i,h} m \quad \text{for bias } x_i \quad (9)
\]
Here \( \overline{u}^{m}_n \) is determined by Eq. (12).

\[
\begin{align*}
\overline{U}^{m}_n &= -F^{m}(n^{m}_n) \\
U^{m}_q &= F(n^{m}_q)(W^{m+1})^{-1}U^{m+1}_q \\
U^{m} &= [U^{m}_1, U^{m}_2, \ldots, U^{m}_q]
\end{align*}
\] (10) (11) (12)

\section{Training and optimizing NNs}

The training speed and prediction accuracy of NNs depend on the number of hidden layers and the number of neurons in the hidden layers. NNs with high training speed and prediction accuracy can be obtained through training and optimization with many training vectors, including the ratio of quantum charge density to classical charge density. The training vectors can be obtained by solving the coupled Schrödinger-Poisson equations self-consistently for MOSFETs with a variety of oxide thicknesses, silicon layer thicknesses, doping levels, and applied gate voltages.

When solving the coupled Schrödinger-Poisson equations, they must be discretized by the finite difference method first. Then the Poisson equation is solved to obtain the classical potential distribution by an iterative method. The potential is used to solve the Schrödinger equation along the direction vertical to the gate. The new charge density can be calculated with wavefunctions and energy levels obtained from the Schrödinger equation. The new charge density is plugged into the Poisson equation to solve the new potential. The Schrödinger equation is solved again with the new potential. These steps are repeated iteratively until the convergence criterion is met.

The oxide thickness of MOSFETs used for training and optimizing NNs varies from 1 to 5nm, the silicon layer thickness varies from 3 to 100nm, the applied gate voltage ranges from 0.5 to 1.5V, and the doping concentration varies from \(1 \times 10^{15}\) to \(5 \times 10^{18}\) cm\(^{-3}\). The ratio of quantum to classical charge densities at any depth in the silicon layers of MOSFETs is calculated by solving the coupled Schrödinger-Poisson equations.

The charge density of single gate MOSFETs in the silicon layer varies with oxide thickness, silicon layer thickness, gate voltage, depth, and doping level. The ratio is also a function of the five parameters. Therefore, the input layer of the NNs for single gate MOSFETs has five neurons, representing the five parameters. Because the value and varying scope of doping concentration are very large, the logarithm of doping concentration is used in training vectors.

The computer used to train and optimize NNs is equipped with a Pentium 2.2G CPU, 512M memory, and 80G of disk space.

First, NNs with one hidden layer containing different numbers of neurons were built and trained, in which the stopping criterion for MSE was \(10^{-5}\), and the maximum epoch was 1000. The training curves are shown in Fig. 2(a). The numbers at the upper right hand corner of the figure represent the number of neurons in the hidden layer. It can be seen that the MSE of the NNs with only one hidden layer is larger than \(10^{-4}\).

![Fig. 2](image)

Fig. 2 Training curves with different hidden layers (a) NN with one hidden layer; (b) NN with two hidden layers; (c) NN with three hidden layers
the NNs meet the specified stopping criterion of $10^{-5}$ before the maximum epoch of 1000 is reached. Considering the influence of the random initial weights and biases, it can be concluded that the number of neurons in the hidden layers has no evident effect on the training accuracy and speed.

At last, NNs with three hidden layers were set up and trained. There were seven and three neurons in the first and second hidden layers, respectively, but the number of neurons in the third hidden layer was different. The criteria for MSE and the maximum epoch were the same as above. The training curves are plotted in Fig. 2 (c), in which the numbers at the upper right hand corner of the figure are the number of neurons in the third hidden layer. It can be seen that the MSE of the NN with seven neurons in the third hidden layer does not converge to the stopping criterion of $10^{-5}$. In addition, the average training time of NNs with three hidden layers per epoch is 0.057, while the corresponding time of NNs with two hidden layers is only 0.024.

Thus NNs with two hidden layers should be used to obtain high training speed and prediction accuracy.

The electron densities obtained by a trained NN with two hidden layers and Schrödinger-Poisson (SP) approach for two single gate nMOSFETs against depth are shown in Fig. 3(a). The NN has seven and three neurons in the first and second hidden layers, respectively. The doping level of the two MOSFETs is $N_d = 10^{17}$ cm$^{-3}$, the applied gate voltage is 1.5 V, and the oxide thicknesses are 1 and 3 nm, respectively. The average relative differences between the densities by the two methods for the two MOSFETs are 0.4% and 0.3%, respectively.

For two-gate MOSFETs, the input layer of the NNs should have one more neuron representing the second gate voltage. The NNs were trained in the same way as the single gate MOSFETs. It is also found that high training speed and accuracy could be achieved by NNs with two hidden layers and the number of neurons in the hidden layers has no evident effect. The electron densities obtained by a trained NN with two hidden layers and Schrödinger-Poisson approach for a two-gate nMOSFET against depth are shown in Fig. 3(b), in which the doping level $N_d$ is $10^{17}$ cm$^{-3}$, the oxide thickness of both gates is 1 nm, the applied voltages for the front and back gates are 1.5 and 1 V, respectively, and the silicon thickness is 5 nm. The average relative difference between the electron densities by the two methods is 0.5%.

Fig. 3 Electron density by NNs and SP approaches against depth (a) Single gate; (b) Double gate

The capacitances of a 20 μm × 20 μm nMOS capacitor with a 1.6 nm-thick oxide, obtained by the two methods are presented in Fig. 4. The average relative difference between the capacitances is 0.5%.

Fig. 4 Capacitance against gate voltage
4 Conclusion

BP NNs using the Levenberg–Marquardt algorithm can be used to construct a predictive model for the quantum charge density of MOSFETs. High training speed and prediction accuracy can be obtained using the NNs with two hidden layers, but the number of neurons in the hidden layers has no evident effect. Our model can predict the quantum charge densities in the silicon layer of single and double-gate MOSFETs in very good agreement with Schrödinger–Poisson equations. The model can be used in nanoscale MOSFET modeling and simulation.

References


MOSFET

" 