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

Li Zunchao[1,†], Jiang Yaolin[2], and Zhang Ruizhi[1]

(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 back-propagation 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 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 it agrees closely with the Schrödinger-Poisson approach.

Key words: neural network; quantum correction; nanoscale MOSFET; charge density
PACC: 6185; 0300; 7115Q
CLC number: TN304.2 **Document code:** A **Article ID:** 0253-4177(2006)03-0438-05

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 SiO_2/Si interface[1,2] ——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[3,4], 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[5]. 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 SiO_2/Si interface. The Levenberg-Marquardt algorithm is used to avoid these drawbacks[6].

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 SiO_2/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_i^m of neuron i in hidden layer m is

$$n_i^m = w_{i,1}^m a_1^{m-1} + w_{i,2}^m a_2^{m-1} + \cdots + w_{i,S^{m-1}}^m a_{S^{m-1}}^{m-1} + b_i^m \quad (1)$$

where a_1^{m-1}, a_2^{m-1}, $\cdots$, and $a_{S^{m-1}}^{m-1}$ are the outputs of

* Project supported by the National Natural Science Foundation of China (No. 60472003) and the State Key Development Program for Basic Research of China (No. 2005CB321701)
† Corresponding author. Email: zcli@mail.xjtu.edu.cn
Received 22 August 2005, revised manuscript received 20 October 2005

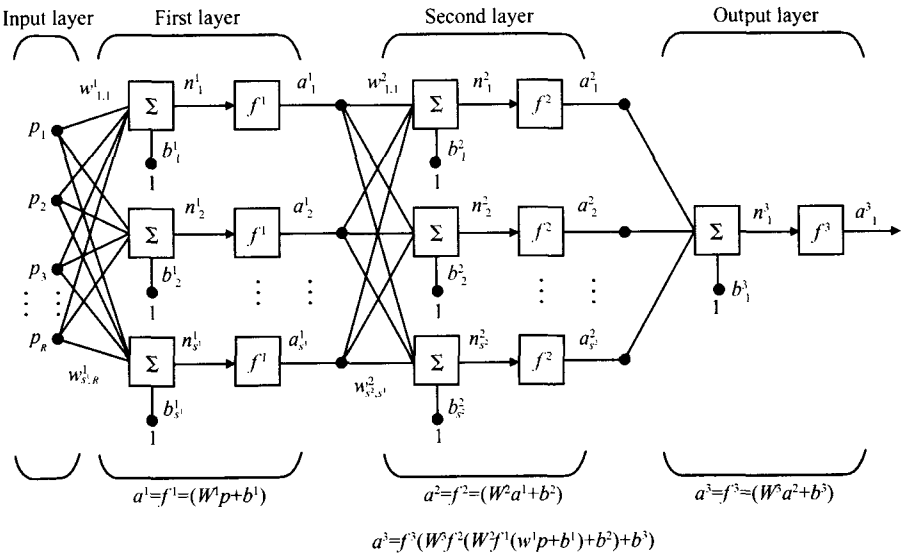

Fig. 1 NN structure

the neurons in hidden layer $m-1$, S^{m-1} is the number of neurons in hidden layer $m-1$, $w^m_{i,j}$ is the weight between neuron j in hidden layer $m-1$ and neuron i in hidden layer m, and b^m_i is the bias of neuron i in hidden layer m. The output a^m_i of neuron i in hidden layer m is

$$a_i^m = f^m(n_i^m) = 2/(1+\exp(-2n_i^m)) - 1 \quad (2)$$

where f^m is the activation function.

The outputs $a^m_1, a^m_2, \cdots, a^m_{S^m}$ 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^O 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

$$\mathrm{MSE} = \frac{\sum_q^Q (t_q - a_q)^2}{Q} = \frac{\sum_q^Q e_q^2}{Q} \quad (3)$$

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 $w^m_{i,j}$ and biases b^m_i corresponding to the highest prediction accuracy.

In the training process, the weight and bias vector x is adjusted by[7]

$$\Delta x_k = -(J^T(x_k)J(x_k) + \mu_k I)^{-1} J^T(x_k)\, e(x_k) \quad (4)$$

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 μ is a scalar quantity used for controlling the search direction and step. e, x, and J are given in Eqs. (5～7), respectively.

$$e^T = [e_1\ e_2\ \cdots\ e_Q] \quad (5)$$

$$x^T = [x_1\ x_2\ \cdots\ x_n] = [w^1_{1,1}\ w^1_{1,2}\ \cdots w^1_{S^1,R} b^1_1\ \cdots\ b^1_{S^1}\ w^2_{1,1}\ \cdots b^M_1] \quad (6)$$

Here R is the number of neurons in the input layer, M denotes the output layer, and b^M_1 is the bias of the neuron in the output layer.

$$J(x) = \begin{vmatrix} \frac{\partial e_1}{\partial w^1_{1,1}} & \frac{\partial e_1}{\partial w^1_{1,2}} & \cdots & \frac{\partial e_1}{\partial w^1_{S^1,R}} & \frac{\partial e_1}{\partial b^1_1} & \cdots & \frac{\partial e_1}{\partial b^M_1} \\ \frac{\partial e_2}{\partial w^1_{1,1}} & \frac{\partial e_2}{\partial w^1_{1,2}} & \cdots & \frac{\partial e_2}{\partial w^1_{S^1,R}} & \frac{\partial e_2}{\partial b^1_1} & \cdots & \frac{\partial e_2}{\partial b^M_1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & & \cdots \\ \frac{\partial e_Q}{\partial w^1_{1,1}} & \frac{\partial e_Q}{\partial w^1_{1,2}} & \cdots & \frac{\partial e_Q}{\partial w^1_{S^1,R}} & \frac{\partial e_Q}{\partial b^1_1} & \cdots & \frac{\partial e_Q}{\partial b^M_1} \end{vmatrix} \quad (7)$$

The elements of J are computed by Eqs. (8) and (9).

$$[J]_{h,l} = \frac{\partial e_h}{\partial x_l} = \frac{\partial e_{k,q}}{\partial w^m_{i,j}} = \overline{u}^m_{i,h} \times a^{m-1}_{j,q}, \quad \text{for weight } x_l \quad (8)$$

$$[J]_{h,l} = \frac{\partial e_h}{\partial x_l} = \frac{\partial e_{k,q}}{\partial b^m_i} = \overline{u}^m_{i,h}, \quad \text{for bias } x_l \quad (9)$$

Here $\bar{u}^{m}_{i,h}$ is determined by Eq. (12).

$$\bar{U}^{M}_{q} = -\dot{F}^{M}(n^{M}_{q}) \tag{10}$$

$$\bar{U}^{m}_{q} = \dot{F}(n^{m}_{q})(W^{m+1})^{T}\bar{U}^{m+1}_{q} \tag{11}$$

$$\bar{U}^{m} = [\bar{U}^{m}_{1} | \bar{U}^{m}_{2} | \cdots | \bar{U}^{m}_{Q}] \tag{12}$$

3 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[8~10]. 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[11~14].

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 1×10^{15} to 5×10^{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}.

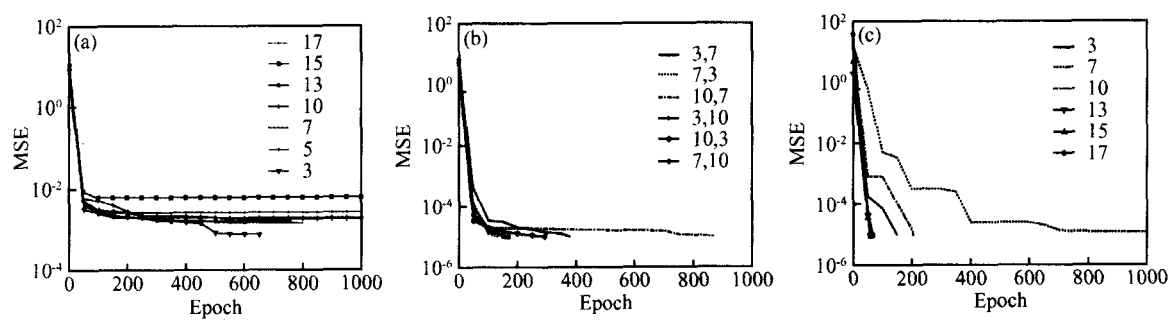

Fig. 2 Training curves with different hidden layers (a) NN with one hidden layer; (b) NN with two hidden layers; (c) NN with three hidden layers

Then NNs with two hidden layers containing different numbers of neurons were constructed and trained, in which the criteria for the MSE and maximum epoch were the same as above. The training curves are shown in Fig. 2(b). The numbers at the upper right hand corner of the figure are the number of neurons in the first and second hidden layers, respectively. It can be seen that the MSEs of all

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 $N_a = 10^{17}\,cm^{-3}$, the applied gate voltage is 1.5V, and the oxide thicknesses are 1 and 3nm, 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_a is $10^{17}\,cm^{-3}$, the oxide thickness of both gates is 1nm, the applied voltages for the front and back gates are 1.5 and 1V, respectively, and the silicon thickness is 5nm. The average relative difference between the electron densities by the two methods is 0.5%.

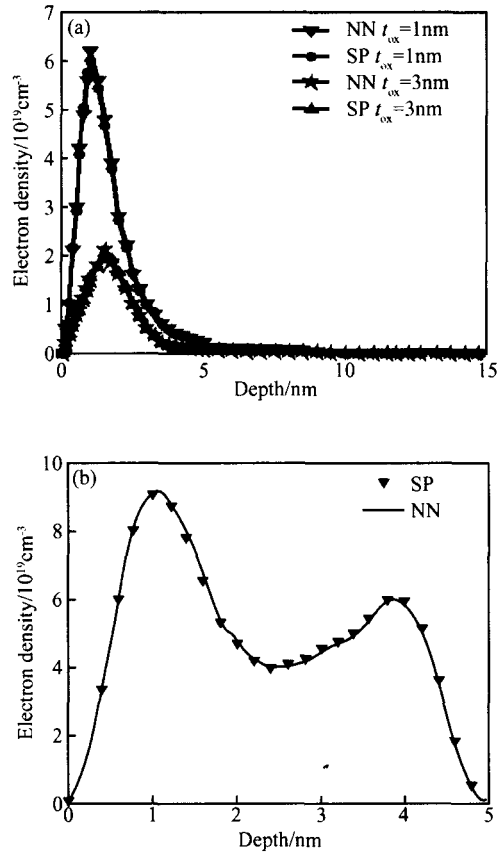

Fig. 3 Electron density by NNs and SP approaches against depth (a) Single gate; (b) Double gate

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

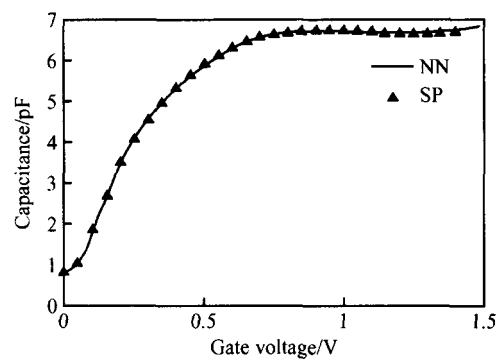

Fig. 4 Capacitance against gate voltage

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

[1] Gan Xuewen, Huang Ru, Liu Xiaoyan, et al. Nanoscale CMOS devices. Beijing: Science Press, 2004 (in Chinese) [甘学温, 黄如, 刘晓彦, 等. 纳米 CMOS 器件. 北京: 科学出版社, 2004]

[2] Wigger S J. Modeling ultra-small semiconductor devices. Arizona State University, 2002

[3] Choi C. Modeling of nanoscale MOSFETs. Stanford University, 2002

[4] Xia T S, Register L F, Banerjee S K. Quantum transport in double-gate MOSFETs with complex band structure. IEEE Trans Electron Devices, 2003, 50(6): 1511

[5] Byunghwhan K, Sungmo K, Lee D W. Predictive model of a reduced surface field p-LDMOSFET using neural network. Solid-State Electron, 2004, 48: 2153

[6] Hatami S, Azizi M Y, Bahrami H R. Accurate and efficient modeling of SOI MOSFET with technology independent neural networks. IEEE Trans Computer-Aided Design of Integrated Circuits and Systems, 2004, 23(11): 1580

[7] Martin T H, Howard B D, Mark B. Neural network design. Beijing: China Machine Press, 2002

[8] Tang T W, Li Y. A SPICE-compatible model for nanoscale MOSFET capacitor simulation under the inversion condition. Nanotechnology, 2002, 1: 243

[9] Wang X, Tang T W. Comparison of three quantum correction models for the charge density in MOS inversion layers. J Comput Electron, 2002, (1): 283

[10] Li Y, Tang T, Wang X. Modeling of quantum effects for ultrathin oxide MOS structures with an effective potential. IEEE Trans Nanotechnol, 2002, (1): 238

[11] Li Y M, Chao T S, Sze S M. A novel parallel approach for quantum effect simulation in semiconductor devices. Int J Modeling Simulation, 2003, 23: 94

[12] Fu Ying, Lu Wei. Physics of semiconductor quantum device. Beijing: Science Press, 2005 (in Chinese) [傅英, 陆卫. 半导体量子器件物理. 北京: 科学出版社, 2005]

[13] Chen W Q, Register L F, Banerjee S K. Simulation of quantum effects along the channel of ultrascaled Si-based MOSFETs. IEEE Trans Electron Devices, 2002, 49(4): 652

[14] Mudanai S, Register L F, Tasch A F, et al. Understanding the effects of wave function penetration on the inversion layer capacitance of nMOSFETs. IEEE Electron Device Lett, 2001, 22(3): 145

基于神经网络的纳米 MOSFET 载流子密度量子更正 *

李尊朝[1,†] 蒋耀林[2] 张瑞智[1]

(1 西安交通大学电子与信息工程学院, 西安 710049)

(2 西安交通大学理学院, 西安 710049)

摘要: 为了处理纳米 MOSFET 载流子分布的量子效应, 提出了基于 Levenberg-Marquardt BP 神经网络的量子更正模型, 通过载流子的经典密度计算其量子密度, 并对拥有不同隐层数和隐层神经元数的神经网络的训练速度和精度进行了研究. 结果表明: 含有 2 个隐层的神经网络具有高的训练速度和精度, 但隐层神经元数对速度和精度的影响并不明显; 对于单栅和双栅纳米 MOSFET, 其载流子量子密度可以通过神经网络进行快速计算, 其结果与 Schrödinger-Poisson 方程的吻合程度很高.

关键词: 神经网络; 量子更正; 纳米 MOSFET; 电荷密度

PACC: 6185; 0300; 7115Q

中图分类号: TN304.2 **文献标识码**: A **文章编号**: 0253-4177(2006)03-0438-05

* 国家自然科学基金(批准号: 60472003)和国家重点基础研究发展计划(批准号: 2005CB321701)资助项目

† 通信作者. Email: zcli@mail.xjtu.edu.cn

2005-08-22 收到, 2005-10-20 定稿