Efficient Initialization of Artificial Neural Network Weights for Electrical Component Models

Tuomo Kujanpää and Janne Roos

Helsinki University of Technology, Department of Electrical and Communications Engineering, Circuit Theory Laboratory, P.O.Box 3000, FI-02015 TKK, Finland.
tuomo.kujanpaa@tkk.fi
janne@ct.tkk.fi

1 Introduction

The modeling of RF/microwave components for computer-aided design is facing new challenges because of increasing operation frequencies, circuit complexity, integration density, and decreasing time to market. Recently, it has been shown that Artificial Neural Networks (ANNs) offer solutions to urgent modeling problems encountered with conventional numerical methods (e.g., 3-D EM simulation) and empirical models. Fast and accurate models based on ANNs have been created for a wide range of components [ZG00k], [PAR01].
The crucial part in ANN-based modeling is ANN training, that is, optimization of ANN weights with given measurements or, say, 3-D EM simulation data. In [TF97] several ANN weight-initialization methods were introduced and compared mainly by means of classification problems. It was shown how the choice of an initialization method influences the convergence of the optimization and the optimal initial weights are, by some means, determined by the measurement/simulation data set. However, weight-initialization methods have not previously been systematically evaluated for electrical component modeling problems and the nature of the problems — the functions to be approximated — differs significantly from, e.g., classification problems with discrete/Boolean input/target values.
In this paper, three methods for an initialization of ANN weights are experimentally evaluated for electrical component modeling applications. The third method, a special modification of the second method, is not found in literature. The methods are evaluated with respect to average ANN training error, ANN test error, and ANN training CPU time. Also, the standard deviations of ANN training and test errors are calculated for robustness analysis of the methods.

2 Artificial neural networks

The most widely used ANN in the field of RF/microwave component modeling is the Multi-Layer Perceptron (MLP) [ZG00k]. The three-layer MLP used in this work realizes the nonlinear mapping

$$\tilde{y}_l(\mathbf{x}, \mathbf{w}) = w_{l0} + \sum_{j=1}^{N_\mathrm{h}} w_{lj}\, a\tanh\left(b \cdot (w_{j0} + \sum_{i=1}^{N_\mathrm{i}} w_{ji} x_i) \right), \qquad (1)$$

$$l = 1, 2, \ldots, N_\mathrm{o},$$

where N_i, N_h, and N_o represent the number of inputs, hidden-layer neurons, and outputs, respectively; $\mathbf{x} = (x_1, x_2, \ldots, x_{N_\mathrm{i}})$, $\tilde{\mathbf{y}} = (\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_{N_\mathrm{o}})$, and $\mathbf{w} = (w_{10}, w_{11}, \ldots, w_{N_\mathrm{o} N_\mathrm{h}})$

represents ANN inputs, outputs, and weights, respectively. The function $a\tanh(bv_j)$ is called the Activation Function (AF), where the parameters a and b determine the maxima and the steepness, respectively, and $v_j = w_{j0} + \sum_{i=1}^{N_\mathrm{i}} w_{ji}x_i$ is the induced local field of the function. Let $\mathbf{y} = \mathbf{y}(\mathbf{x})$ be an unknown, nonlinear, multidimensional function to be approximated by the MLP mapping (1): $\tilde{\mathbf{y}} = \tilde{\mathbf{y}}(\mathbf{x}, \mathbf{w})$. Let $\{(\mathbf{x}^k, \mathbf{y}^k), k = 1, 2, \ldots, N_\mathrm{tr}\}$ be an appropriate training set, N_tr being the number of samples, and the training-set inputs and targets being scaled linearly in the range $[-1, 1]$. Furthermore, let us define the normalized training error as

$$E_\mathrm{tr}(\mathbf{w}) = \sqrt{\frac{1}{N_\mathrm{tr}N_\mathrm{o}} \sum_{k=1}^{N_\mathrm{tr}} \sum_{l=1}^{N_\mathrm{o}} \left(\frac{\tilde{y}_l(\mathbf{x}^k, \mathbf{w}) - y_l^k}{2} \right)^2}. \tag{2}$$

The training of the ANN means minimizing $E_\mathrm{tr}(\mathbf{w})$ with respect to the weights, $\mathbf{w}$, using a suitable optimization method — in this work, Hestenes–Stiefel conjugate-gradient with Error Back Propagation (EBP) [KRH05k]. The generalization capability of the trained ANN is evaluated by applying Eq. (2) to an independent test set, $\{(\mathbf{x}^k, \mathbf{y}^k), k = 1, 2, \ldots, N_\mathrm{te}\}$, to obtain the normalized test error $E_\mathrm{te}(\mathbf{w})$.

3 Weight-initialization methods

Weight initialization tries to provide initial weight values close to the global minimum of $E_\mathrm{tr}(\mathbf{w})$, in the hope of avoiding local minima. There are several strategies for initializing the MLP weights; the most developed strategies can also be regarded as training methods [EFP05]. However, the most widely utilized strategy for ANN-based RF/microwave component modeling is, still, initializing the weights as random real numbers from a Uniform Distribution (UD) with fixed or variable range. The weight-initialization Methods (Ms) evaluated in this work include: M1. random initialization from UD with fixed range [ZG00k], M2. random initialization from UD with variable range and special input data scaling [Hay99k], and M3. random initialization from UD with variable range and special input and target training data scaling.
Utilizing M1 [ZG00k], one sets $a = b = 1$ and $w_{ji}, w_{lj} \in [-c, c]$, where, e.g., $c = 1.0$. This heuristic initialization tries to ensure the local field (v_j) of the AFs to be such that it forces the AFs to operate in an approximately linear transition region determined by maxima of the second derivative, $\max(\partial^2 \tanh(v_j)/\partial v_j^2)$. This would be desirable for the convergence of optimization because, when using EBP [Hay99k], $\partial E_\mathrm{tr}^2/\partial w_{ji} \sim \partial \tanh(v_j)/\partial v_j$ and the latter has its maximum value in the transition region. However, the heuristic weight initialization does not take into account the mean, $\bar{x}_i$, and the standard deviation of input data, σ_{x_i}, and, therefore, AFs may operate in saturation regions slowing down the optimization [Hay99k].
M2 [Hay99k] forces the specific AFs to operate in the transition region (between (–1,–1) and (1,1) for $a = 1.7159$ and $b = 2/3$) with $w_{ji} \in [-\sqrt{3/N_\mathrm{i}}, \sqrt{3/N_\mathrm{i}}]$, $w_{lj} \in [-\sqrt{3/N_\mathrm{h}}, \sqrt{3/N_\mathrm{h}}]$, $w_{j0} = 0$, and $w_{l0} = 0$. This initialization is based on a special input data scaling, with $\bar{x}_i = 0$ and $\sigma_{x_i} = 1$.
When one utilizes M2 and approximates the transition region of AFs as a straight line going through the origin with slope 1, the distribution parameters of the MLP outputs, $\tilde{y}_l$, are $\bar{\tilde{y}}_l = 0$ and $\sigma_{\tilde{y}_l} = 1$ as for MLP inputs x_i. A hypothesis to be tested is presented (M3): scaling of the target training data, y_l, such that $\bar{y}_l = 0$ and $\sigma_{y_l} = 1$, improves the convergence of optimization. The idea of M3 is to equalize the distribution parameters of the MLP outputs and the target training data, possibly aiding the convergence.

4 Experimental setup

In the evaluation, we had eight representative modeling problems: 1. approximation of a modulated sinusoidal function, 2. the same problem with additive normal-distributed noise, 3.

MEMS gas-damper behavior, 4. rounded-stripline-bend parallel capacitance and series inductance vs. device geometries, 5. JFET DC characteristics, 6. spiral-inductor S-parameters vs. geometries, 7. power amplifier output power vs. supply voltage and frequency, and 8. MESFET drain and gate currents vs. bias voltages and temperature.
For each problem, three appropriately sized MLPs (N_h and N_w get three different values as given in Table 1) were utilized. The modeling-problem characterization and corresponding MLPs are shown in Table 1, where N_w is the resulting number of ANN weights, i.e., optimization variables, N_tr is the number of training-set samples, and $N_\mathrm{g} = N_\mathrm{tr}N_\mathrm{o}$ is the resulting number of optimization goals.

Table 1: Modeling-problem characterization

problem	N_i	N_h	N_o	N_tr	N_w	N_g
1	1	{5,10,15}	1	20	{16,31,46}	20
2	1	{5,10,15}	1	20	{16,31,46}	20
3	3	{5,10,15}	1	40	{26,51,76}	40
4	3	{5,10,15}	2	50	{32,62,92}	100
5	2	{5,10,20}	3	306	{33,63,123}	918
6	5	{10,15,25}	5	486	{115,170,280}	2430
7	2	{10,20,30}	1	4667	{41,81,121}	4667
8	3	{10,15,25}	2	37597	{62,92,152}	75194

Each MLP was trained 30 times with each weight-initialization method — M1 with $c = 0.001, 0.005, 0.01, 0.1, 0.5, 1.0, 5.0$ — and E_tr, E_te, and training CPU time noted in hundred-step increments. The results obtained for each method were averaged over all runs at each value of the optimization cycles. In addition, the standard deviations of the training and test errors were calculated for each problem and method. Finally, the standard deviations were averaged over all the problems at each value of the optimization cycles.
A total number of 6480 runs were carried out by semi-automatic scripts using APLAC 8.2 `ANNModelGenerator` [A06k] on an Ia64 HP Server rx5670 with a 1.3 GHz processor and 4 Gbyte memory.

5 Analysis of results

A set of representative results is shown in Figs. 1–5. The convergence of M1 degraded rapidly with increasing or decreasing c (as in [TF97]), and therefore only the best results (obtained with $c = 0.5$) for M1 are shown.
According to the results obtained, the hypothesis presented is true; comparing the new M3 to M1 (with $c = 0.5$), the training and test errors decreased by 13.6 % and 1.4 %, respectively. The smallest standard deviations for training and test errors show that M3 is also more robust than other methods (41.6 % and 2.1 % improvement, respectively, compared to M1 with $c = 0.5$). The performance improvement is obtained with a slight increase in the training CPU time (7.0 % increase compared to M1 with $c = 0.5$).
M2 forces the AFs to operate in the transition region and improves the convergence when compared to heuristic M1 with other values of c. Thus, one can conclude that when $a = b = 1$, $c = 0.5$, and the training-set inputs and targets are scaled linearly in the range $[-1, 1]$, the AFs are forced, on the average, to operate in the transition region. However, this may not be true with a single modeling problem.

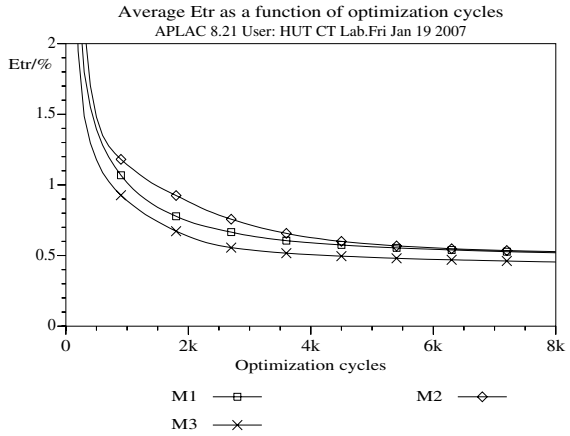

Fig. 1: Average training error vs. optimization cycles

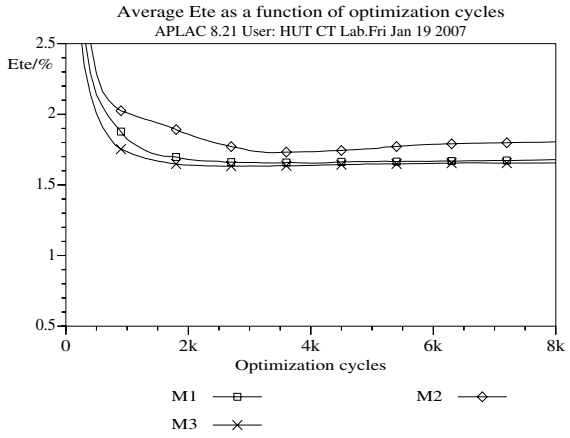

Fig. 2: Average test error vs. optimization cycles

6 Conclusions

Three methods for the initialization of MLP-ANN weights were experimentally evaluated for electrical component modeling applications. A new weight-initialization method was also presented. The methods were evaluated with respect to average training error, test error and training CPU time. Also, the standard deviations of training and test errors were calculated and utilized to analyze robustness of the methods.

According to the results obtained, the hypothesis presented is true: the new method proposed (M3) improves the convergence and robustness of MLP-ANN training for electrical component modeling problems. The performance is improved because the AFs are forced to operate in the transition region and the target training data is scaled so that its distribution parameters correspond to the ones of the MLP outputs. This is not true with the heuristic weight initialization (M1), even though it is possible to find empirically a good value of c for a specific modeling problem.

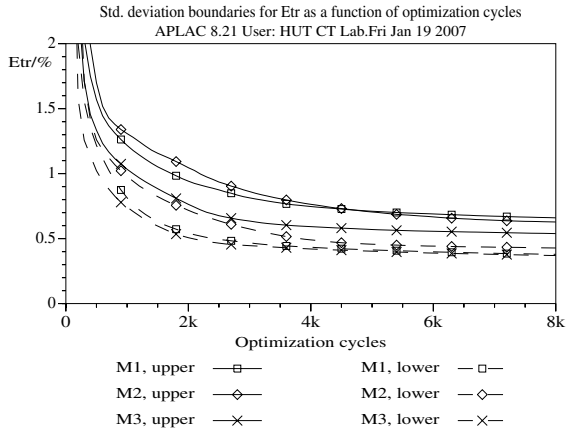

Fig. 3: Standard deviation boundaries for training error vs. optimization cycles

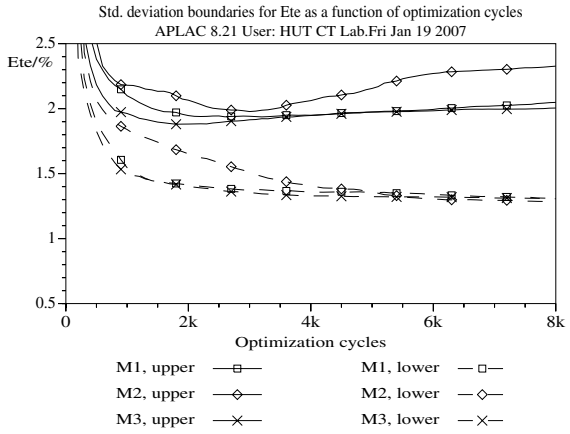

Fig. 4: Standard deviation boundaries for test error vs. optimization cycles

Acknowledgment

This work was funded by Nokia Corporation and AWR-APLAC Corporation through projects TEKES/ELMO/MOSAICS (grants 2078/31/03 and 2440/31/03) and TEKES/MASI/AMAZE (grant 3239/31/05).

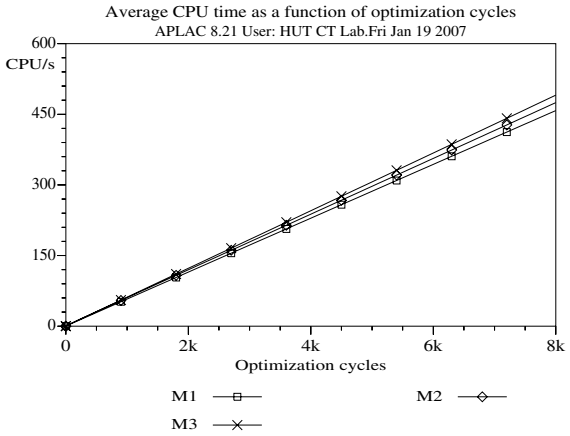

Fig. 5: Average training CPU time vs. optimization cycles

References

[ZG00k] Zhang, Q.J., Gupta, K.C.: Neural Networks for RF and Microwave Design. Artech House, Boston London (2000)

[PAR01] Plebe, A., Anile, A.M., Rinaudo, S.: Sub-micrometer bipolar transistor modeling using neural networks. In: van Rienen, U., Günther, M., Hecht, D. (Eds.): Scientific Computing in Electrical Engineering, Lecture Notes in Computational Science and Engineering, **18**, Springer, Berlin Heidelberg, 259–266 (2001)

[TF97] Thimm, G., Fiesler, E.: High-order and multilayer perceptron initialization. IEEE Trans. on Neural Networks, **2**, 349–359 (1997)

[KRH05k] Kujanpää, T., Roos, J., Honkala, M.: Experimental comparison of optimization methods in ANN training. In: Proc. PRIME 2005, **2**, 430–433 (2005)

[EFP05] Erdogmus, D., Fontenla-Romero, O., Principe, J. C., Alonso-Betanzos, A.: Linear-least-squares initialization of multilayer perceptrons through backpropagation of the desired response. IEEE Trans. on Neural Networks, **2**, 325–337 (2005)

[Hay99k] Haykin, S.: Neural Networks — A Comprehensive Foundation. Prentice Hall, New Jersey (1999)

[A06k] APLAC 8.2 Manuals. AWR-APLAC Corporation (2006)