View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Publications at Bielefeld University

Tolerance of Radial Basis Functions against Stuck-At-Faults*

Ralf Eickhoff[1] and Ulrich Rückert[1]

Heinz Nixdorf Institute
System and Circuit Technology
University of Paderborn, Germany
eickhoff,rueckert@hni.upb.de

Abstract. Neural networks are intended to be used in future nanoelectronic systems since neural architectures seem to be robust against malfunctioning elements and noise in their weights. In this paper we analyze the fault-tolerance of Radial Basis Function networks to Stuck-At-Faults at the trained weights and at the output of neurons. Moreover, we determine upper bounds on the mean square error arising from these faults.

1 Introduction

Neural networks are used as function approximators for continuous functions [1, 2]. Especially, Radial Basis Function networks are utilized to perform a local approximation of an unknown function specified by a set of test data. The main reason why neural networks are used for this purpose is the adaptability of the network due to the learning process. Moreover, the networks seem to be fault-tolerant against malfunctioning neurons [2] which can be modeled as Stuck-At faults and to be robust against noise corrupted weights and inputs [3].

Digital and analog implementations of neural networks have always to face malfunctioning elements [4], especially in future nanoelectronic realizations [5]. Moreover, when using analog hardware noise is always present due to thermal or flicker noise [6–8] and even if digital hardware is used quantization noise contaminates the weights and inputs [9]. Thus, the artificial neural network structure should handle malfunctioning elements and noise contaminated weights.

In this paper we analyze the Radial Basis Function network with respect to errors based on Stuck-At-Faults. In [10] these properties are demonstrated for sigmoidal feedforward networks. First, a short overview about the analyzed neural network architecture is given. The fault-tolerance against different types of Stuck-At-Faults is analyzed afterwards. Section 4 determines upper bounds on the mean square error for Stuck-At-Faults occuring in the weights and output of neurons and necessary restrictions are introduced leading to bounded errors.

*This work was supported by the Graduate College 776 - Automatic Configuration in Open Systems- funded by the Deutsche Forschungsgemeinschaft.

2 Radial Basis Functions

In this section a short overview about the architecture of a Radial Basis Function network is given. The network consists of an input vector with dimension $\dim \boldsymbol{x} = n$. At a second step m different Basis Functions which have different centers $\boldsymbol{x}_i$ are superposed and denoted by a weight α_i to produce the output.

The Radial Basis Function network (RBF) can be used for local function approximation [11]. Basing on the regularization theory the quadratic error is minimized with respect to a stabilizing term [12]. Due to this stabilizer the interpolation and approximation quality is controlled in order to achieve a smooth approximation. Based on this stabilizer different Basis Functions can be performed for superposition. As a consequence, the network function can be expressed as

$$f_m(\boldsymbol{x}) = \sum_{i=1}^{m} \alpha_i h_i \left(\|\boldsymbol{x} - \boldsymbol{x}_i\| \right) \tag{1}$$

where m denotes the number of superposed Basis Functions.

The function $h_i(z)$ can be any function related to a (radial) regularization stabilizer. Here, the stabilizer leading to a Gaussian function is considered, thus it follows

$$h_i(z) = \exp\left(\frac{-z^2}{2\sigma_i^2} \right) \tag{2}$$

and therefore

$$f_m(\boldsymbol{x}) = \sum_{i=1}^{m} \alpha_i \exp\left(\frac{-\|\boldsymbol{x} - \boldsymbol{x}_i\|^2}{2\sigma_i^2} \right) \tag{3}$$

Moreover, the parameters $\boldsymbol{x}_i$ are the individual centers of each Basis Function, σ_i^2 resembles the variance of each Gaussian function and α_i are the weights from each neuron to the output neuron, which performs a linear superposition of all Basis Functions.

3 Stuck-At-Faults in Radial Basis Function networks

In future nanoelectronic systems one major problem will be the massive amount of malfunctioning elements [5]. Therefore, fault-tolerant architectures have to be established in order to achieve reliable systems and predictable system behavior. From biology it is well known that the human brain allows the loss of several neurons because of the redundancy in its structure [2, 13]. However, it was proven that sigmoidal feedforward neural networks are not fault-tolerant against Stuck-At-Faults [10, 14]. Here, the fault-tolerance of an RBF network is analyzed against Stuck-At-Faults at the output weights, at the Basis Function centers and at the variance of the Gaussian Basis Function.

3.1 Stuck-At-Fault at the output weight

First, it is assumed that the Stuck-At-Fault occurs in the weights from the neuron to the output α_i. In order to achieve a universal expression it is assumed that the weight is sticking at the value μ. Moreover, with loss of generality only one weight from the whole Basis Functions is imposed by this fault. Thus, it follows for the difference of both network outputs

$$f_m(\boldsymbol{x}) - \hat{f}_m(\boldsymbol{x}) = \sum_{i=1}^{m} \alpha_i \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2\sigma_i^2}\right) - \sum_{i=1}^{m} \hat{\alpha}_i \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2\sigma_i^2}\right) \tag{4}$$

$$= \sum_{i=1}^{m} (\alpha_i - \hat{\alpha}_i) \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2\sigma_i^2}\right) \tag{5}$$

where $\hat{f}_m(\boldsymbol{x})$ denotes the network function due to the Stuck-At-Fault at the k-th neuron. Since nearly all α_i are identical to $\hat{\alpha_i}$ the difference vanishes except of the k-th term where the weight is sticking at the value μ. Therefore, under the assumption that only one weights is imposed by a Stuck-At-Fault (5) leads to

$$f_m(\boldsymbol{x}) - \hat{f}_m(\boldsymbol{x}) = (\alpha_k - \mu) \cdot \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_k\|^2}{2\sigma_k^2}\right) \tag{6}$$

3.2 Stuck-At-Fault at an RBF center

Here, the Stuck-At-Fault occurs in the center of a Basis Function resulting in an unintentional movement of the center. The k-th neuron is interfered by a Stuck-At-Fault at the ν-th entry of the vector $\boldsymbol{x}_k$, and therefore this produces a faulty output behavior of the neural network. The difference between both network responses can be expressed

$$f_m(\boldsymbol{x}) - \hat{f}_m(\boldsymbol{x}) = \sum_{i=1}^{m} \alpha_i \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2\sigma_i^2}\right) - \sum_{i=1}^{m} \alpha_i \exp\left(-\frac{\|\boldsymbol{x}-\hat{\boldsymbol{x}}_i\|^2}{2\sigma_i^2}\right) \tag{7}$$

$$= \alpha_k \left[\exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_k\|^2}{2\sigma_k^2}\right) - \exp\left(-\frac{\|\boldsymbol{x}-\hat{\boldsymbol{x}}_k\|^2}{2\sigma_k^2}\right)\right] \tag{8}$$

$$= \alpha_k \exp\left(-\frac{\sum_{\substack{j=1\\ j\neq\nu}}^{n} (x_j - x_{kj})^2}{2\sigma_k^2}\right) \times \tag{9}$$

$$\left[\exp\left(-\frac{(x_\nu - x_{k\nu})^2}{2\sigma_k^2}\right) - \exp\left(-\frac{(x_\nu - \mu)^2}{2\sigma_k^2}\right)\right] \tag{10}$$

where x_{kj} denotes the j-th entry in the center of the k-th Basis Function.

3.3 Stuck-At-Fault at the variance of a Gaussian Basis Function

Now, it is assumed that the variance of a certain Basis Function is affected by a Stuck-At-Fault at μ. Here, the k-th Basis Function is disturbed, leading to

$$f_m(\boldsymbol{x}) - \hat{f}_m(\boldsymbol{x}) = \sum_{i=1}^{m} \alpha_i \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2\sigma_i^2}\right) - \sum_{i=1}^{m} \alpha_i \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2\hat{\sigma}_i^2}\right) \quad (11)$$

$$= \alpha_k \cdot \left[\exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_k\|^2}{2\sigma_k^2}\right) - \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_k\|^2}{2\mu^2}\right)\right] \quad (12)$$

4 Bounds on the mean square error

In this section we analyze the fault-tolerance of an RBF network against the Stuck-At-Faults. Hence, for the three different types of Stuck-At-Faults necessary restrictions are introduced to achieve an upper bound on the mean square error of the difference between both network functions. Therefore, the input vector is assumed to be a random variable with a certain distribution function. In the following E denotes the expected value.

Concerning (6), the mean square error due to a Stuck-At-Fault at the output weights is determined by

$$\mathrm{mse}_\alpha = (\alpha_k - \mu)^2 \cdot E\Bigg[\underbrace{\exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_k\|^2}{\sigma_k^2}\right)}_{\leq 1}\Bigg] \quad (13)$$

Equation (13) can be estimated by the mean value theorem of integral calculus [15] resulting in a mean square error of

$$\mathrm{mse}_\alpha \leq (\alpha_k - \mu)^2 \quad (14)$$

Thus, if the weights of the RBF network are not bounded rather arbitrary the mean square has no upper bound. Moreover, the error is depending on the Stuck-At value. The value of the Stuck-At-Fault is a consequence of the technical implementation. In analog hardware μ can be restricted to any continuous value in a certain interval which is determined by operating conditions [6, 8]. In the case of a digital realization μ can only adopt discrete values leading to a quantized error. However, in both implementations μ is restricted by an upper bound and with restricted weights

$$|\alpha_i| \leq B \quad \forall\, i = 1 \ldots n \quad (15)$$

equation (14) can be further evaluated

$$\mathrm{mse}_\alpha \leq (B + |\mu|)^2 \quad (16)$$

In the same way equation (10) and (12) can be determined leading to

$$\begin{aligned}
\mathrm{mse}_{\boldsymbol{x}_i} &= \alpha_k^2 E\left[\left(\exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2\sigma_i^2}\right) - \exp\left(-\frac{\|\boldsymbol{x}-\hat{\boldsymbol{x}}_i\|^2}{2\sigma_i^2}\right)\right)^2\right] \\
&\leq \alpha_k^2 \qquad (17) \\
\mathrm{mse}_{\sigma} &= \alpha_k^2 E\left[\left(\exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2\sigma_i^2}\right) - \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2\mu^2}\right)\right)^2\right] \\
&\leq \alpha_k^2 \qquad (18)
\end{aligned}$$

From equation (17) and (18) can be concluded that both mean square errors have no upper bound since the weights of the neural network can be arbitrary. In contrast to (14) the mean square errors do not depend on the technical realization. If the weights are restricted by an upper bound $|\alpha| \leq B$ both errors are restricted by an upper bound

$$\mathrm{mse} \leq B^2 \qquad (19)$$

5 Conclusion

Artificial neural networks are intended to be fault-tolerant against noise contaminated inputs and malfunctioning elements like biological neural networks. However, it was shown in [10, 14] that sigmoidal feedforward networks are not fault-tolerant. In this work the fault-tolerance against malfunctioning elements is determined for Radial Basis Function networks. These interferences can be modeled as Stuck-At-Faults at the output weights and at the output behavior of the neurons.

As in the case for multilayer perceptrons the Radial Basis Function network is not immune to malfunctioning elements. If arbitrary weights can be used no upper bound on the mean square error exists. Therefore, a well-defined system behavior due to sticking elements can not be guaranteed. Furthermore, if the error occurs in the output weights of a neuron the mean square error is depending on the sticking value and thus on the technical realization.

The technical implementation of neural networks in analog or digital hardware provides restrictions on the weights which are resulting in fault-tolerant networks. As the weights are bounded by an upper bound (15) the mean square error is restricted (cf. (16) and (19)). In the case of analog hardware the Stuck-At-Fault can be assigned to any continuous value within a certain interval determined by the operating conditions. For digital implementations the Stuck-At-Fault are restricted to '1' and '0' leading to quantized steps of the error.

However, both technical implementations provide upper bounds on the Stuck-At-Faults as was shown in section 4. Therefore, technical realizations of an RBF network are still fault-tolerant against malfunctioning elements. By providing adequate bounds on the weights a reliable network response can be guaranteed.

References

1. Geva, S., Sitte, J.: A constructive method for multivariate function approximation by multilayer perceptrons. IEEE Transactions on Neural Networks **3** (1992) 621–624
2. Haykin, S.: Neural Networks. A Comprehensive Foundation. Second edn. Prentice Hall, New Jersey, USA (1999)
3. Rückert, U., Surmann, H.: Tolerance of a binary associative memory towards stuck-at-faults. In Kohonen, T., ed.: Artificial Neural Networks. Volume 2., Amsterdam, North-Holland (1991) 1195–1198
4. Rückert, U., Kreuzer, I., Tryba, V.: Fault-tolerance of associative memories based on neural networks. In: Proceedings of the International Conference on Computer Technology, Systems and Applications, Hamburg, Germany (1989) 1.52–1.55
5. Beiu, V., Rückert, U., Roy, S., Nyathi, J.: On nanoelectronic architectural challenges and solutions. Fourth IEEE Conference on Nanotechnology (2004)
6. Razavi, B.: Design of Analog CMOS Integrated Circuits. McGraw-Hill (2000)
7. Sitte, J., Körner, T., Rückert, U.: Local cluster neural net: Analog vlsi design. Neurocomputing **19** (1998) 185 – 197
8. Körner, T., Rückert, U., Geva, S., Malmstrom, K., Sitte, J.: Vlsi friendly neural network with localied transfer functions. In: Proceedings of the IEEE International Conference on Neural Networks. Volume 1., Perth, Australia (1995) 169 – 174
9. Widrow, B., Kollár, J.: Quantization Noise. Prentice Hall PTR, New Jersey, USA (2002)
10. Chandra, P., Singh, Y.: Feedforward sigmoidal networks - equicontinuiy and fault-tolerance properties. IEEE Transactions on Neural Networks **15** (2004) 1350–1366
11. Girosi, F., Poggio, T.: Networks and the best approximation property. Biological Cybernetics **63** (1990) 169–176
12. Girosi, F., Jones, M., Poggio, T.: Regularization theory and neural networks architectures. Neural Computation **7** (1995) 219–269
13. Bose, N.K., Liang, P.: Neural network fundamentals with graphs, algorithms, and applications. McGraw-Hill, Inc. (1996)
14. Phatak, D.S., Koren, I.: Complete and partial fault tolerance of feedforward neural nets. IEEE Transactions on Neural Networks **6** (1995) 446–456
15. Bronstein, I.N., Semendyayev, K.A.: Handbook of mathematics (3rd ed.). Springer-Verlag (1997)