ESANN'2006 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 26-28 April 2006, d-side publi., ISBN 2-930307-06-4.

Robust Local Cluster Neural Networks

Ralf Eickhoff[1] and Joaquin Sitte[2] and Ulrich Rückert[1] *

1- Heinz Nixdorf Institute - System and Circuit Technology
University of Paderborn - Germany

2- Smart Devices Lab - School of Computing Science and Software Engineering
Queensland University of Technology - Australia

Abstract. Artificial neural networks are intended to be used in future nanoelectronics since their biological examples seem to be robust to noise. In this paper, we analyze the robustness of Local Cluster Neural Networks and determine upper bounds on the mean square error for noise contaminated weights and inputs.

1 Introduction

Neural networks are used for function approximation of any continuous functions [1, 2]. Especially, basis function networks are utilized since their response function has mathematically well defined behavior. However, multilayer perceptrons are intended to model biology and, therefore, it is assumed that the characteristic of robustness is adopted by its artificial equivalents [3, 4].

Robust architectures will be needed in future technology trends because information processing will become more susceptible to noise due to shrinking dimensions and signal amplitudes. Estimating the impact of noisy inputs and parameters is essential to establish robust systems operating in noisy environments. In this work, the equicontinuity and robustness of the Local Cluster Neural Networks, a multilayer perceptron approximating local basis functions, is analyzed.

2 Local Cluster Neural Network (LCNN)

Here, we give only a brief summary of the LCNN. A detailed description of the architecture can be found in [1]. The idea of the LCNN is to combine sigmoid functions in such a way that they only respond to a finite region in the input space. This is accomplished by first constructing a *ridge function* for every input signal, by pairing two sigmoids. The ridge function can be described as

$$l(\vec{w}, \vec{r}, k_1, \vec{x}) = \sigma(k_1, \vec{w}^T(\vec{x} - \vec{r}) + 1) - \sigma(k_1, \vec{w}^T(\vec{x} - \vec{r}) - 1) \quad (1)$$

The orientation and width of the ridge is determined by the orientation of $\vec{w}$ and its length, respectively. The position of the ridge is given by the position vector $\vec{r}$. The sigmoid is chosen to be the logistic function with steepness k_1

$$\sigma(k_1, x) = \frac{1}{1 + \mathrm{e}^{-k_1 x}} \quad (2)$$

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

A local function is obtained by adding the ridge functions of all the components in the input signal vector. All ridges have a different orientation, but the same center

$$f\left(\vec{W},\vec{r},k_1,\vec{x}\right)=\sum_{i=1}^{n} l(\vec{w}_i,\vec{r},k_1,\vec{x}) \tag{3}$$

where $\vec{W}$ is a $n\times n$ matrix made out of n ridge (column) vectors $\vec{w}_i$. This function has a bump around the common center $\vec{r}$ and ridges emanating to infinity in as many directions as there are input dimensions. These ridges have to be removed to make the function local. Application of a properly biased sigmoid σ_o to the function $f\left(\vec{W},\vec{r},k_1,\vec{x}\right)$ cuts off the ridges smoothly

$$\mathcal{L}\left(\vec{W},\vec{r},k_2,\vec{x}\right)=\sigma_o\left(f\left(\vec{W},\vec{r},k_1,\vec{x}\right)-b\right) \tag{4}$$

A local cluster network consists of an array of local cluster functions all receiving the same inputs. The network output is a weighted sum of the local cluster outputs and describes a set of functions

$$\mathcal{S}=\left\{y_m(\vec{x})\middle|y_m(\vec{x})=\sum_{\nu=1}^{m}\alpha_\nu\mathcal{L}\left(\vec{W}_\nu,\vec{r}_\nu,\vec{x}\right),m\in I\!N,\vec{r}_\nu\in I\!R^n,\vec{W}_\nu\in I\!R^{n\times n}\right\} \tag{5}$$

3 Equicontinuity and Robustness

The equicontinuous property of a function set is an important feature to produce a stable approximation which means that two slightly different inputs produce only slightly different output behavior. Therefore, only small errors occur at the output if the inputs are contaminated by noise and, furthermore, the nonequicontinuity results in an unstable approximation meaning that large discrepancies are possible while providing nearly identical values at the input.

Consequently, if the property of equicontinuity is fulfilled by a neural network the whole network is expected to be noise immune. If the inputs or weights are contaminated with noise the network response will differ only slightly in contrast to the ideal output behavior. The equicontinuous property is defined as [5, 6]

Definition 1 *Let χ be a compact metric space with metric d, and let $\mathcal{S}$ be a nonempty subset of $\mathcal{C}(\chi)$. By definition, if f is a member of $\mathcal{S}$ then f is continuous, that is, for each $\epsilon>0$, there exist $\delta>0$ such that $d(x,y)<\delta\Rightarrow|f(x)-f(y)|<\epsilon$. $\mathcal{S}$ is said to be equicontinuous if for each ϵ a $\delta(\epsilon)$ can be found that serves at once for all functions f in $\mathcal{S}$.*

3.1 Input Space

To prove the equicontinuous property in the input space the difference between two slightly different input vectors have to be determined

$$|y_m(\vec{x})-y_m(\vec{y})|=\left|\sum_{\nu=1}^{m}\alpha_\nu\mathcal{L}\left(\vec{W}_\nu,\vec{r}_\nu,\vec{x}\right)-\sum_{\nu=1}^{m}\alpha_\nu\mathcal{L}\left(\vec{W}_\nu,\vec{r}_\nu,\vec{y}\right)\right| \tag{6}$$

Since (6) is differentiable the mean value theorem can be applied

$$|y_m(\vec{x}) - y_m(\vec{y})| = \left|\nabla y_m\left(\vec{\xi}\right) \cdot (\vec{x} - \vec{y})\right| \tag{7}$$

where the gradient can be determined as

$$\begin{aligned} \frac{\partial y_m(\vec{x})}{\partial x_k} &= \sum_{i=1}^{m} \alpha_i \frac{\partial \mathcal{L}_i(\vec{x})}{\partial x_k} = \sum_{i=1}^{m} \alpha_i \sigma_o' \left(f(\vec{x}) - b\right) \frac{\partial f(\vec{x})}{\partial x_k} \\ &= \sum_{i=1}^{m} \alpha_i \sigma_o'(\cdot) \sum_{j=1}^{n} \frac{\partial l_{ij}}{\partial x_k} = \sum_{i=1}^{m} \alpha_i \sigma_o'(\cdot) \sum_{j=1}^{n} w_{ijn} \left(\sigma'(\cdot) - \sigma'(\cdot)\right) \end{aligned} \tag{8}$$

and (8) is the k-th component of the gradient.

Furthermore, the maximum of the derivative of both sigmoids can be evaluated as

$$\lim_{x \to 0} \sigma'(x) = \frac{1}{4} k_1, \quad \lim_{x \to 0} \sigma_o'(x) = \frac{1}{2} k_2 \tag{9}$$

Using the mean value theorem, the triangle inequality and the maximum of the sigmoids equation (7) can be evaluated

$$\begin{aligned} |y_m(\vec{x}) - y_m(\vec{y})| &\le \sum_{\nu=1}^{n} \left| \sum_{i=1}^{m} \alpha_i \sigma_o'(\cdot) \sum_{j=1}^{n} w_{ij\nu} \left(\sigma'(\cdot) - \sigma'(\cdot)\right) \right| d(\vec{x}, \vec{y}) && (10) \\ &\le \sum_{\nu=1}^{n} \left| \sum_{i=1}^{m} \frac{k_2 \alpha_i}{2} \sum_{j=1}^{n} \frac{k_1}{4} w_{ij\nu} \right| d(\vec{x}, \vec{y}) && (11) \\ &\le \frac{k_1 k_2}{8} \sum_{\nu=1}^{n} \sum_{i=1}^{m} \sum_{j=1}^{n} |\alpha_i w_{ij\nu}| \, d(\vec{x}, \vec{y}) = \epsilon && (12) \\ \Rightarrow \delta &= \frac{8\epsilon}{k_1 k_2 \sum_{\nu=1}^{n} \sum_{i=1}^{m} \sum_{j=1}^{n} |\alpha_i w_{ij\nu}|} && (13) \end{aligned}$$

As a consequence of (12) the equicontinuous property is not valid for the LCNN in the input space, since relation (12) must be consistent to Definition 1 for all functions of set $\mathcal{S}$ defined in (5). As a result of equation (12) and (13) the parameter ϵ and δ are depending on the weights α_ν and $w_{ij\nu}$. Again, these weights are based on the function which should be approximated by the neural network and, consequently, both parameters (ϵ, δ) are not independent of the function. Thus, as the MLP and the RBF network the LCNN has not the equicontinuous property in the input space [5, 7].

But, if the weights are constrained by

$$|w_{ij\nu}| \le T \quad \wedge \quad |\alpha_\nu| \le B \tag{14}$$

equation (12) reduces to

$$\begin{aligned} \left|y_m\left(\vec{x}\right) - y_m\left(\vec{y}\right)\right| &\leq \frac{k_1 k_2}{8} \sum_{\nu=1}^{n} \sum_{i=1}^{m} \sum_{j=1}^{n} B \cdot T \cdot d\left(\vec{x}, \vec{y}\right) && (15) \\ &= \frac{k_1 k_2}{8} n^2 m B T = \epsilon && (16) \\ \Rightarrow \delta &= \frac{8\epsilon}{k_1 k_2 n^2 m B T} && (17) \end{aligned}$$

Consequently, the LCNN achieves the equicontinuous property in the input space if the constraints of (14) are provided. Therefore, the mean square error (mse) can be calculated if it is assumed that the inputs are contaminated with Gaussian noise with finite variance. The mse can be evaluated as

$$\text{mse} \leq \frac{k_1^2 k_2^2}{64} \left(\sum_{\nu=1}^{n} \sum_{i=1}^{m} \sum_{j=1}^{n} \left|\alpha_i w_{ij\nu}\right| \right)^2 n\sigma_n^2 \tag{18}$$

where n identical noise sources (inputs) contribute to a noisy output and σ_n^2 denotes their variance.

Constraining the weights gives an upper bound to the mse

$$\text{mse} \leq \frac{k_1^2 k_2^2}{64} n^4 m^2 B^2 T^2 n \sigma_n^2 = \frac{k_1^2 k_2^2}{64} n^5 m^2 B^2 T^2 \sigma_n^2 \tag{19}$$

3.2 Parameter Space

Let assume that the weight space is endowed with a metric, $\mathcal{W}_1$ and $\mathcal{W}_2$ are two distinct points in the weight space and $d\left(\mathcal{W}_1, \mathcal{W}_2\right)$ denotes their distance. Then the equicontinuous property can be written as

$$\left|y_m\left(\mathcal{W}^1\right) - y_m\left(\mathcal{W}^2\right)\right| \leq \left(\left|\nabla_w y_m(\vec{\xi})\right| + \left|\nabla_{\vec{r}} y_m(\vec{\xi})\right| + \left|\nabla_\alpha y_m(\vec{\xi})\right| \right) d\left(\mathcal{W}^1, \mathcal{W}^2\right) \tag{20}$$

where ∇_w denotes the gradient with respect to the weights, $\nabla_{\vec{r}}$ with respect to the centers and ∇_α with respect to the output weights.

Each gradient can be determined

$$\begin{aligned} \left|\nabla_w y_m(\vec{\xi})\right| &= \sum_{\mu=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} \left| \alpha_\mu \frac{\partial \mathcal{L}_\mu}{\partial w_{\mu ij}} \right| && (21) \\ &\leq \frac{k_1 k_2}{8} \sum_{\mu=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} \left|\alpha_\mu\right| \left|x_j - r_{\mu j}\right| && (22) \\ &= \frac{k_1 k_2}{8} \sum_{\mu=1}^{m} n \alpha_\mu \|\vec{x} - \vec{r}_\mu\|_1 && (23) \end{aligned}$$

and

$$\left|\nabla_{\vec{r}} y_m(\vec{\xi})\right| \leq \sum_{\mu=1}^{m}\sum_{j=1}^{n} |\alpha_\mu| \left|\frac{\partial \mathcal{L}_\mu}{\partial r_{\mu j}}\right| \tag{24}$$

$$\leq \sum_{\mu=1}^{m}\sum_{j=1}^{n} \frac{|\alpha_\mu| k_1 k_2}{8} \sum_{i=1}^{n} |w_{\mu ij}| \tag{25}$$

$$= \frac{k_1 k_2}{8} \sum_{\mu=1}^{m} |\alpha_\mu| \sum_{j=1}^{n}\sum_{i=1}^{n} |w_{\mu ij}| \tag{26}$$

and also

$$\left|\nabla_{\alpha} y_m(\vec{\xi})\right| = \sum_{\mu=1}^{m} |\mathcal{L}_\mu| \leq \sum_{\mu=1}^{m} 1 = m \tag{27}$$

From applying (23), (26) and (27) to (20) it follows for the equicontinuity in the weight space

$$\left|y_m\left(\mathcal{W}^1\right) - y_m\left(\mathcal{W}^2\right)\right| \leq \left(\frac{k_1 k_2}{8} \sum_{\mu=1}^{m} n\alpha_\mu \|\vec{x} - \vec{r}_\mu\|_1 + \frac{k_1 k_2}{8} \sum_{\mu=1}^{m} |\alpha_\mu| \sum_{j=1}^{n}\sum_{i=1}^{n} |w_{\mu ij}| + m\right) d\left(\mathcal{W}^1, \mathcal{W}^2\right) = \epsilon \tag{28}$$

and δ can be determined as

$$\delta = \frac{\epsilon}{\frac{k_1 k_2}{8} \sum_{\mu=1}^{m} n\alpha_\mu \|\vec{x} - \vec{r}_\mu\|_1 + \frac{k_1 k_2}{8} \sum_{\mu=1}^{m} |\alpha_\mu| \sum_{j=1}^{n}\sum_{i=1}^{n} |w_{\mu ij}| + m} \tag{29}$$

As a consequence of (28) the LCNN does not have the equicontinuous property in the weight space since the parameters ϵ and δ depend on the weights α_μ and $w_{\mu ij}$ and, moreover, the Manhattan distance between the input vector and the centers. Since these weights are dependent on the actual function which has to be approximated ϵ and δ also depend on the function. Thus, the LCNN is not equicontinuous in the weight space.

Providing the constraints of (14) and also restricting the Manhattan distance

$$\|\vec{x} - \vec{r}_\mu\|_1 \leq C \tag{30}$$

equation (28) can be further evaluated

$$\left|y_m(\mathcal{W}^1) - y_m(\mathcal{W}^2)\right| \leq \left(\frac{k_1 k_2}{8} mnBC + \frac{k_1 k_2}{8} mn^2 BT + m\right) d(\mathcal{W}^1, \mathcal{W}^2) \tag{31}$$

respectively

$$\delta = \frac{\epsilon}{\frac{k_1 k_2}{8} m \cdot n \cdot B \cdot C + \frac{k_1 k_2}{8} m \cdot n^2 \cdot B \cdot T + m} \tag{32}$$

and achieving an equicontinuous network in the weight space.

Furthermore, the mse can be determined if Gaussian noise contaminates all parameters of the network, and this leads to

$$\text{mse} \leq \left[\frac{k_1 k_2}{8} \left(\sum_{\mu=1}^{m} n\alpha_\mu \|\vec{x} - \vec{r}_\mu\|_1 + \sum_{\mu=1}^{m} |\alpha_\mu| \sum_{j=1}^{n} \sum_{i=1}^{n} |w_{\mu ij}| \right) + m \right]^2 m(n^2+n+1)\sigma_n^2 \tag{33}$$

where, at all, $m \cdot (n^2 + n + 1)$ parameters contribute to a noisy output.

By constraining all the parameters (14) and also the Manhattan distance (30) the mse has an upper bound

$$\text{mse} \leq \left(\frac{k_1 k_2}{8} m \cdot n \cdot B \cdot C + \frac{k_1 k_2}{8} m \cdot n^2 \cdot B \cdot T + m \right)^2 m(n^2 + n + 1)\sigma_n^2 \tag{34}$$

4 Conclusion

We have shown the equicontinuity and robustness of the LCNN under certain restrictions. Moreover, upper bounds on the mean square error are determined for noise contaminated inputs and weights.

Without these restrictions the LCNN is not equicontinuous and not noise immune. But restricting the trained weights by an upper resp. lower bound and the Manhattan distance between the input vectors and the centers this leads to a noise immune network. Fortunately, technical implementations provide these bounds naturally due to their limited resolution or memory and its number representation. But as a result of the limited resolution of digital systems quantization noise contaminates the inputs and weights whereas analog systems have to face thermal and flicker noise. However, the network response can be determined and guaranteed by our analysis.

References

[1] Shlomo Geva and Joaquin Sitte. A constructive method for multivariate function approximation by multilayer perceptrons. *IEEE Transactions on Neural Networks*, 3(4):621–624, July 1992.

[2] Simon Haykin. *Neural Networks. A Comprehensive Foundation*. Prentice Hall, New Jersey, USA, second edition, 1999.

[3] R. J. MacGregor. *Neural Modeling: Electrical Signal Processing in the Nervous System*. Plenum Pub Corp, 1977.

[4] Michael A. Arbib. *The Handbook of Brain Theory and Neural Networks: Second Edition*. The MIT Press, Cambridge, MA, USA, 2002.

[5] Pravin Chandra and Yogesh Singh. Feedforward sigmoidal networks - equicontinuiy and fault-tolerance properties. *IEEE Transactions on Neural Networks*, 15(6):1350–1366, 2004.

[6] G. F. Simmons. *Topology and Modern Analysis*. McGray-Hill, Tokyo, Japan, 1964.

[7] Ralf Eickhoff and Ulrich Rückert. Robustness of radial basis functions. In *Proceedings of the 8th International Work-Conference on Artificial Neural Networks (IWANN)*, pages 264–271, Barcelona, Spain, 8 - 10 June 2005.