Honam Mathematical J. **34** (2012), No. 3, pp. 341–349
http://dx.doi.org/10.5831/HMJ.2012.34.3.341

CONSTRUCTIVE APPROXIMATION BY GAUSSIAN NEURAL NETWORKS

Nahmwoo Hahm† and Bum Il Hong‡

Abstract. In this paper, we discuss a constructive approximation by Gaussian neural networks. We show that it is possible to construct Gaussian neural networks with integer weights that approximate arbitrarily well for functions in $C_c(\mathbb{R}^s)$. We demonstrate numerical experiments to support our theoretical results.

1. Introduction

In recent years, a great deal of research in the theory and the application of neural networks has been done by many researchers ([1], [2], [3], [4], [7], [10]).

Neural networks estimate functions that are linear combinations of activation functions composed of affine functionals. The following functions are typically used as an activation function.

$$\sigma(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \qquad \text{(The Heaviside function)}$$

$$\sigma(x) = 1/(1+e^{-x}) \qquad \text{(The squashing function).}$$

Note that the Heaviside function and the squashing function are sigmoidal functions.

Definition 1.1. *A sigmoidal function is a function $\sigma : \mathbb{R} \to \mathbb{R}$ such that*

$$\lim_{t\to\infty} \sigma(t) = 1 \quad \textit{and} \quad \lim_{t\to-\infty} \sigma(t) = 0.$$

Received May 21, 2012. Accepted June 5, 2012.
2000 Mathematics Subject Classification. 41A25, 41A46, 42A85.
Key words and phrases. Constructive Approximation, Neural Network, Gaussian Activation Function.
† This research was supported by the University of Incheon Research Fund, 2011.
‡ Corresponding author.

We basically investigate the approximation by a neural network with one hidden layer. This neural network has an input layer, a hidden layer and an output layer. A neural network with one hidden layer is of the form

$$\sum_{1\le \mathbf{i}\le n} p_{\mathbf{i}}\sigma(q_{\mathbf{i}}\mathbf{x}+\mathbf{r}_{\mathbf{i}}) \tag{1.1}$$

where σ is an activation function, $p_{\mathbf{i}}, q_{\mathbf{i}} \in \mathbb{R}$, and $\mathbf{x}, \mathbf{r}_{\mathbf{i}} \in \mathbb{R}^s$. Note that $q_{\mathbf{i}}'s$ and $\mathbf{r}_{\mathbf{i}}'s$ are called weights and thresholds, respectively.

Chen [2] and Hahm and Hong [6] showed approximation orders to functions in $C[0,1]$ and $\bar{C}(\mathbb{R})$ by neural networks with a sigmoidal activation function, respectively. Although their proofs are constructive, the approximation algorithms in [2] and [6] are not applicable in $C[0,1]^s$ and $\bar{C}(\mathbb{R}^s)$ when $s \ge 2$ is an integer. In addition, we are not able to use the constructive proofs in [2] and [6] if the activation function in a neural network is not a sigmoidal function.

Recently, some researchers showed the importance of Gaussian neural networks. Beheral, Gopal and Chaudhury [1] used Gaussian neural networks for robot tracking applications to model both the forward and the inverse dynamics of a robot arm. Firmin, Hamand, Postaire and Zhang [4] showed the efficiency of Gaussian neural networks to fault detection in glass bottle inspection. Hartman, Keeler and Kowalski [5] theoretically proved that neural network with a single layer of hidden units of gaussian type is a universal approximator for real-valued functions on convex and compact sets of $\mathbb{R}^s$ but their proofs are not constructive. Theoretically, Mhaskar [9] showed an approximation order to functions in Sobolev space by neural networks with a Gaussian activation function but Mhaskar did not suggest explicit weights and thresholds.

In this research, we suggest explicit weights and thresholds of Gaussian neural networks and demonstrate numerical experiments to make our theoretical results strong and solid.

2. Preliminaries

Let s be a natural number. For $\mathbf{x} = (x_1, \dots, x_s)$, $\mathbf{y} = (y_1, \dots, y_s) \in \mathbb{R}^s$ and $c \in \mathbb{R}$, we write $\mathbf{x}+\mathbf{y} = (x_1+y_1, x_2+y_2, \dots, x_s+y_s)$ and $c\mathbf{x} = (cx_1, cx_2, \dots, cx_s)$. For $\mathbf{i} = (i_1, i_2, \dots, i_s) \in \mathbb{N}^s$, we define $1 \le \mathbf{i} \le n$ by $1 \le i_1, i_2, \dots, i_s \le n$ and denote a point $\mathbf{b}_{\mathbf{i}} = (b_{i_1}, b_{i_2}, \dots, b_{i_s}) \in \mathbb{R}^s$.

In addition, we define $\|\mathbf{x}\| := \|\mathbf{x}\|_2 = (\sum_{i=1}^{s} |x_i|^2)^{1/2}$. For a function f defined on $A \subset \mathbb{R}^s$, we define

$$\|f\|_{\infty,A} := \sup\{|f(\mathbf{x})| : \mathbf{x} \in A\}.$$

We denote by $C_c(\mathbb{R}^s)$ the set of all continuous functions that have compact supports. Note that

$$\sigma(\mathbf{x}) = \frac{1}{(\sqrt{2\pi})^s} e^{-\|\mathbf{x}\|^2}$$

is the Gaussian function on $\mathbb{R}^s$ and has the following property.

$$\begin{aligned} &\int_{\mathbb{R}^s} \frac{1}{(\sqrt{2\pi})^s} e^{-\|\mathbf{x}\|^2} d\mathbf{x} \\ = &\ \frac{1}{(\sqrt{2\pi})^s} \int_{\mathbb{R}} \cdots \int_{\mathbb{R}} e^{-{x_1}^2} \cdots e^{-{x_s}^2} dx_1 \cdots dx_s \\ = &\ 1. \end{aligned} \tag{2.1}$$

For a positive real number p, we define $\sigma_p(\mathbf{x}) = p^s \sigma(p\mathbf{x})$ for $\mathbf{x} \in \mathbb{R}^s$. Then, by integration by substitution,

$$\int_{\mathbb{R}^s} \sigma_p(\mathbf{x}) d\mathbf{x} = \int_{\mathbb{R}^s} p^s \sigma(p\mathbf{x}) d\mathbf{x} = \int_{\mathbb{R}^s} \sigma(\mathbf{x}) d\mathbf{x} = 1. \tag{2.2}$$

Let $\epsilon > 0$ be given. Since $\sigma \in L^1(\mathbb{R}^s)$, there exists $B \subset \mathbb{R}^s$ such that

$$\int_{\mathbb{R}^s - B} \sigma(\mathbf{x}) d\mathbf{x} < \epsilon. \tag{2.3}$$

If $f, g \in C(\mathbb{R}^s)$ with $\mathrm{supp}(f) \subset [-a, a]^s$ for some positive real number a, then the convolution of f and g is defined by

$$(f * g)(\mathbf{x}) = \int_{\mathbb{R}^s} f(\mathbf{x} - \mathbf{y}) g(\mathbf{y}) d\mathbf{y}.$$

Note that $(f * g)(\mathbf{x}) = (g * f)(\mathbf{x})$ for any $\mathbf{x} \in \mathbb{R}^s$.

3. Main Results

In the neural network approximation, researchers first use polynomials to approximate target functions and then approximate polynomials by neural networks. In this paper, we use a convolution method [8] in order to obtain explicit coefficients, weights and thresholds.

Theorem 3.1. *Let σ be the Gaussian function and let $\sigma_m(\mathbf{x}) = m^s\sigma(m\mathbf{x})$ for $m \in \mathbf{N}$ and $\mathbf{x} \in \mathbb{R}^s$. For $f \in C_c(\mathbb{R}^s)$ and $\epsilon > 0$, there exists $m_0 \in \mathbb{N}$ such that*

$$\|f - f * \sigma_{m_0}\|_{\infty,\mathbb{R}^s} < \epsilon.$$

Proof. By (2.2), we get

$$\begin{aligned} |f(\mathbf{x}) - (f * \sigma_m)(\mathbf{x})| &= |f(\mathbf{x}) - \int_{\mathbb{R}^s} f(\mathbf{y} - \mathbf{x})\sigma_m(\mathbf{y})d\mathbf{y}| \\ &= \Big|\int_{\mathbb{R}^s} \big(f(\mathbf{x}) - f(\mathbf{y} - \mathbf{x})\big)\sigma_m(\mathbf{y})d\mathbf{y}\Big| \\ &\le \int_{\mathbb{R}^s} \big|f(\mathbf{x}) - f(\mathbf{y} - \mathbf{x})\big|\sigma_m(\mathbf{y})d\mathbf{y}. \end{aligned} \tag{3.1}$$

Let $\epsilon > 0$ be given. Note that f is uniformly continuous on $\mathbb{R}^s$ since $f \in C_c(\mathbb{R}^s)$. Hence, there exists $\delta > 0$ such that for any $y \in \mathbb{R}^s$ with $\|\mathbf{y}\| < \delta$,

$$|f(\mathbf{x}) - f(\mathbf{y} - \mathbf{x})| < \frac{\epsilon}{2}. \tag{3.2}$$

In order to apply (3.2) to (3.1), we rewrite the last part of (3.1) as

$$\begin{aligned} &\int_{\mathbb{R}^s} \big|f(\mathbf{x}) - f(\mathbf{y} - \mathbf{x})\big|\sigma_m(\mathbf{y})d\mathbf{y} \\ = &\int_{\{\mathbf{y}\in\mathbb{R}^s:\|\mathbf{y}\|<\delta\}} \big|f(\mathbf{x}) - f(\mathbf{y} - \mathbf{x})\big|\sigma_m(\mathbf{y})d\mathbf{y} \\ &+ \int_{\mathbb{R}^s-\{\mathbf{y}\in\mathbb{R}^s:\|\mathbf{y}\|<\delta\}} \big|f(\mathbf{x}) - f(\mathbf{y} - \mathbf{x})\big|\sigma_m(\mathbf{y})d\mathbf{y}. \end{aligned} \tag{3.3}$$

From (2.2) and (3.2), the first part of (3.3) yields that

$$\begin{aligned} &\int_{\{\mathbf{y}\in\mathbb{R}^s:\|\mathbf{y}\|<\delta\}} \big|f(\mathbf{x}) - f(\mathbf{y} - \mathbf{x})\big|\sigma_m(\mathbf{y})d\mathbf{y} \\ < &\ \frac{\epsilon}{2}\int_{\mathbb{R}^s} \sigma_m(\mathbf{y})d\mathbf{y} \\ = &\ \frac{\epsilon}{2}. \end{aligned} \tag{3.4}$$

Note that (3.4) holds for any $m \in \mathbb{N}$.
Since $f \in C_c(\mathbb{R}^s)$, it is clear that $f \in L^1(\mathbb{R}^s)$. By (2.3), there exists $m_0 \in \mathbb{N}$ such that

$$\int_{\mathbb{R}^s-\{\mathbf{x}\in\mathbb{R}^s:\|\mathbf{x}\|<m_0\delta\}} \sigma(\mathbf{x})d\mathbf{x} < \frac{\epsilon}{4\|f\|_{\infty,\mathbb{R}^s}}. \tag{3.5}$$

Hence, the second part of (3.3) yields that

$$
\begin{aligned}
& \int_{\mathbb{R}^s - \{\mathbf{y}\in\mathbb{R}^s : \|\mathbf{y}\| < \delta\}} \big|f(\mathbf{x}) - f(\mathbf{y}-\mathbf{x})\big| \sigma_{m_0}(\mathbf{y}) d\mathbf{y} \\
\leq \;& 2\|f\|_\infty \int_{\mathbb{R}^s - \{\mathbf{y}\in\mathbb{R}^s : \|\mathbf{y}\| < m_0\delta\}} \sigma(\mathbf{y}) d\mathbf{y} \qquad (3.6)\\
< \;& \frac{\epsilon}{2}.
\end{aligned}
$$

Therefore, the combination of (3.4) and (3.6) gives

$$\|f - f * \sigma_{m_0}\|_{\infty, \mathbb{R}^s} < \epsilon.$$

Thus, we complete the proof. □

Now, we approximate $f * \sigma_{m_0}$ by a Gaussian neural network. In the proof, we use the Riemann sum for integral.

Theorem 3.2. *Assume that $f \in C_c(\mathbb{R}^s)$. Then for a given $m \in \mathbb{N}$ and $\epsilon > 0$, there exists a neural network*

$$N_{m,n,s} = \sum_{1 \leq \mathbf{i} \leq n} c_{\mathbf{i}} \sigma(m\mathbf{x} + \mathbf{b_i}) \qquad (3.7)$$

such that

$$\|f * \sigma_m - N_{m,n,s}\|_{\infty, \mathbb{R}^s} < \epsilon, \qquad (3.8)$$

where $c_{\mathbf{i}} \in \mathbb{R}$ and $\mathbf{x}, \mathbf{b_i} \in \mathbb{R}^s$ for $1 \leq \mathbf{i} \leq n$.

Proof. Since $f \in C_c(\mathbb{R}^s)$, we may assume that $\mathrm{supp}(f) \subset [-a, a]^s$ for some positive real number a. Then the support of $f * \sigma_m$ is a subset of $[-2a, 2a]^s$. Hence,

$$
\begin{aligned}
(f * \sigma_m)(\mathbf{x}) &= \int_{\mathbb{R}^s} \sigma_m(\mathbf{y}) f(\mathbf{x} - \mathbf{y}) d\mathbf{y} \\
&= \int_{[-2a,2a]^s} \sigma_m(\mathbf{y}) f(\mathbf{x} - \mathbf{y}) d\mathbf{y} \qquad (3.9)\\
&= \int_{[-2a,2a]^s} \sigma_m(\mathbf{x} - \mathbf{y}) f(\mathbf{y}) d\mathbf{y}.
\end{aligned}
$$

For $n \in \mathbb{N}$ and $\mathbf{i} = (i_1, i_2, \ldots, i_s) \in \mathbb{N}^s$ with $1 \leq \mathbf{i} \leq n$, we set

$$\mathbf{a_i} = (-2a + 4a\frac{i_1}{n}, -2a + 4a\frac{i_2}{n}, \ldots, -2a + 4a\frac{i_s}{n}).$$

Note that the set $\{\mathbf{a_i} = (-2a + 4a\frac{i_1}{n}, -2a + 4a\frac{i_2}{n}, \ldots, -2a + 4a\frac{i_s}{n}) \in \mathbb{R}^s : 1 \leq \mathbf{i} \leq n\}$ is the collection of all grid points of $[-2a, 2a]^s$ that does not contain any point on each axis $x_j = -2a$ for $1 \le j \le s$. Therefore, the total number of grid points of $[-2a, 2a]^s$ we use is n^s. Then

$$N_{m,n,s}(\mathbf{x}) := \sum_{1 \le \mathbf{i} \le n} f(\mathbf{a_i})(\frac{4a}{n})^s \sigma(m\mathbf{x} - m\mathbf{a_i}) \tag{3.10}$$

denotes the Riemann sum of $f * \sigma_m$ on $[-2a, 2a]^s$. Hence,

$$\lim_{n \to \infty} N_{m,n,s}(\mathbf{x}) = (f * \sigma_m)(\mathbf{x})$$

for all $\mathbf{x} \in [-2a, 2a]^s$. By the construction, we can easily see that $(f * \sigma_m)(\mathbf{x}) = 0$ and $N_{m,n,s}(\mathbf{x}) = 0$ for $\mathbf{x} \in \mathbb{R}^s - [-2a, 2a]^s$. Therefore, for a given $\epsilon > 0$ and $m \in \mathbb{N}$, there exists $n \in \mathbb{N}$ such that

$$\|f * \sigma_m - N_{m,n,s}\|_{\infty,\mathbb{R}^s} = \|f * \sigma_m - N_{m,n,s}\|_{\infty,[-2a,2a]^s} < \epsilon.$$

□

From Theorem 3.1 and Theorem 3.2, we obtain the following that is the main theorem of this paper.

Theorem 3.3. *If $f \in C_c(\mathbb{R}^s)$ and σ is the Gaussian function, then for a given $\epsilon > 0$, there exists a neural network*

$$N_{m,n,s} = \sum_{1 \le \mathbf{i} \le n} c_{\mathbf{i}} \sigma(m\mathbf{x} + \mathbf{b_i}) \tag{3.11}$$

such that

$$\|f - N_{m,n,s}\|_{\infty,\mathbb{R}^s} < \epsilon, \tag{3.12}$$

where $c_{\mathbf{i}} \in \mathbb{R}$ and $\mathbf{x}, \mathbf{b_i} \in \mathbb{R}^s$ for $1 \le \mathbf{i} \le n$.

Proof. By Theorem 3.1, there exists $m \in \mathbb{N}$ such that

$$\|f - f * \sigma_m\|_{\infty,\mathbb{R}^s} < \frac{\epsilon}{2}. \tag{3.13}$$

And by Theorem 3.2, there exists a neural network $N_{m,n,s}$ such that

$$\|f * \sigma_m - N_{m,n,s}\|_{\infty,\mathbb{R}^s} < \frac{\epsilon}{2}. \tag{3.14}$$

Therefore, we have

$$\begin{aligned} & \|f - N_{m,n,s}\|_{\infty,\mathbb{R}^s} \\ \le \; & \|f - f * \sigma_m\|_{\infty,\mathbb{R}^s} + \|f * \sigma_m - N_{m,n,s}\|_{\infty,\mathbb{R}^s} \\ < \; & \epsilon. \end{aligned} \tag{3.15}$$

This completes the proof. □

4. Numerical Results and Conclusion

Now, we demonstrate numerical data implemented by MATHEMATICA in order to justify our theory. We select

$$f(x) = \begin{cases} \cos x & \text{if } -\dfrac{\pi}{2} \leq x \leq \dfrac{\pi}{2} \\ 0 & \text{if otherwise} \end{cases}$$

as a target function. We choose σ_4 and σ_{10} for convolution and test the approximation of f by $f * \sigma_4$ and $f * \sigma_{10}$, respectively. The results are the followings.

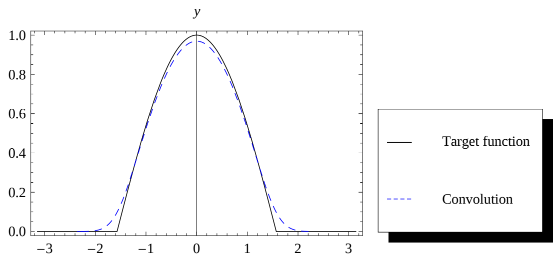

Figure 1. The target function and $f * \sigma_4$.

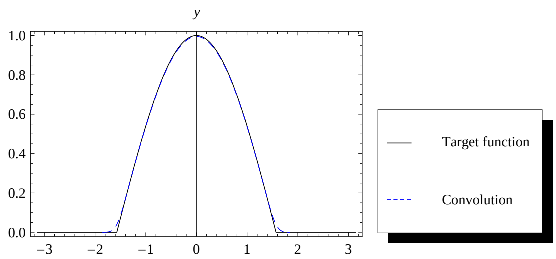

Figure 2. The target function and $f * \sigma_{10}$.

As seen in Figure 1 and Figure 2, $f * \sigma_{10}$ approximates f well on $\mathbb{R}$. In fact, numerical computation shows that $\|f - f * \sigma_4\|_{\infty,\mathbb{R}} = 0.084171$ and $\|f - f * \sigma_{10}\|_{\infty,\mathbb{R}} = 0.013713$.

We then test the approximation of $f * \sigma_{10}$ by $N_{20,10,1}$ and $N_{40,10,1}$, respectively. The results are the followings.

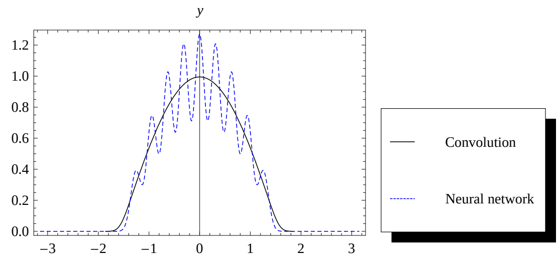

Figure 3. $f * \sigma_{10}$ and neural network $N_{20,20,1}$.

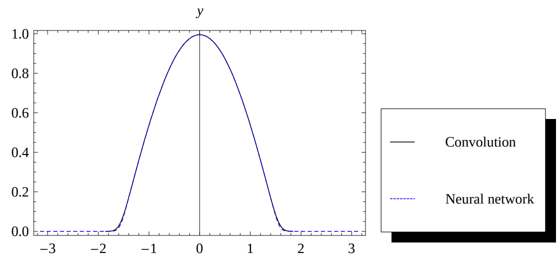

Figure 4. $f * \sigma_{10}$ and neural network $N_{40,20,1}$.

Figure 3 and Figure 4 show that $N_{40,10,1}$ approximates $f * \sigma_{10}$ well on $\mathbb{R}$. In fact, numerical computation shows that $\|f * \sigma_{10} - N_{20,10,1}\|_{\infty,\mathbb{R}} = 0.275447$ and $\|f * \sigma_{10} - N_{40,20,1}\|_{\infty,\mathbb{R}} = 0.00908$.

From these results, we choose a Gaussian neural network $N_{40,20,1}$ to approximate f on $\mathbb{R}$.

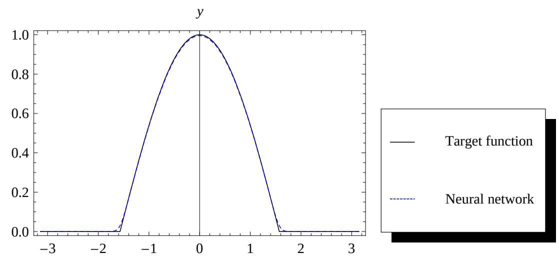

Figure 5. The target function and neural network $N_{40,20,1}$.

Figure 5 shows that $N_{40,10,1}$ approximates f very well on $\mathbb{R}$ and our numerical computation gives $\|f - N_{20,10,1}\|_{\infty,\mathbb{R}} = 0.017826$.

In this research, we have demonstrated the density result for the compact supported continuous functions by neural networks with the Gaussian activation function. Using the convolution method and the

Riemann sum, we have suggested explicit weights and thresholds for neural network approximation, and have obtained the constructive proof for the multivariate functions unlike [2] and [6] in addition.

References

[1] L. Beheral, M. Gopal and S. Chaudhury, *Trajectory tracking of robot manipulator using Gaussian networks*, Robot. Auton. Syst. **13(2)**(1994), 107-115.

[2] D. Chen, *Degree of approximation by superpositions of a sigmoidal function*, Approx. Theory and Appl. **9(3)**(1993), 17-28.

[3] G. Cybenko, *Approximation by superpositions of sigmoidal functions*, Math. Control Signal **2**(1989), 303-314.

[4] C. Firmin, D Hamad, J. Postaire and R. Zhang, *Gaussian neural networks for glass bottles inspection : a learning procedure*, Int. J. Neural Syst. **8(1)**(1997), 41-46.

[5] E. J. Hartman, J. D. Keeler and J. M. Kowalski, *Layered neural networks with Gaussian hidden units as universal approximations*, Neural Comput. **2(2)**(1990), 210-215.

[6] B. I. Hong and N. Hahm, *Approximation order to a function in $\bar{C}(\mathbb{R})$ by superposition of a sigmoidal function*, Appl. Math. Lett. **15**(2002), 591-597.

[7] G. Lewicki and G. Marino, *Approximation of functions of finite variation by superpositions of a sigmoidal function*, Appl. Math. Lett. **17**(2004), 1147-1152.

[8] W. Light, *Techniques for generating approximations via convolution kernels*, Numer. Algorithms **5**(1993), 247-261.

[9] H. N. Mhaskar, *Versatile Gaussian networks*, Proc. IEEE Workshop on Nonlinear Image and Signal Proc. (1995), 70-73.

[10] M. A. Sartori and P. J. Antsaklis, *Gaussian neural networks for control function implementation*, Math. Comput. Model. **23**(1996), 129-142.

Nahmwoo Hahm
Department of Mathematics, University of Incheon,
Incheon 406-772, Korea.
E-mail: nhahm@inchon.ac.kr

Bum Il Hong
Department of Applied Mathematics, Kyung Hee University,
Yongin 446-701, Korea.
E-mail: bihong@khu.ac.kr