CONF-931160--46

10.1 Error Bounds on the Outputs of Artificial Neural Networks

This paper was accepted for presentation at the *American Nuclear Society Winter Meeting*, November 1993, and will be publishe in the proceedings.

Paper removed for separate cycling. CE

FG 02-92 ER 75700

FG 02-92ER 75700

ERROR BOUNDS ON THE OUTPUT OF ARTIFICIAL NEURAL NETWORKS

Eric B. Bartlett and Keehoon Kim

Department of Mechanical Engineering
Nuclear Engineering Program
Iowa State University
Ames, IA 50011

Introduction

Resolving the uncertainties associated with solutions obtained from artificial neural networks (ANNs) is a major concern for ANN researchers[1,2]. Error bounds on the solutions are important because they are an integral part of verification and validation. In this research, stacked generalization[3] (SG) is applied to provide error bounds for novel solutions obtained from ANNs. An outline of SG and its use is given in Figure 1. The data used in this demonstration of SG are given in Figures 2 and 3. This work shows that SG can provide error bounds on ANN results. We have applied SG to nuclear power plant fault detection for verification of diagnoses provided by ANNs[4,5,6].

Error Bound Estimation using Stacked Generalization

Let **F** be an ANN trained on a training set L. Our goal is to estimate error bounds on solutions obtained from **F** for untrained inputs. First, partition the training set L into two subsets, L_1 and L_2. The subset L_2 consists of a single input-output pattern, and L_1 contains the remaining patterns. An ANN **f** having the same architecture as **F** is trained on L_1 and then tested on L_2. Since the pattern in L_2 is not used during training, the ANN output will deviate from the desired output. This deviation represents the ANN's error for an untrained input. The input pattern in L_2 and a vector from the input to its nearest neighbor in L_1 form an input pattern, and the deviation forms an output pattern in a new training set. Another partition is chosen, and the process of training, testing, and obtaining the errors is repeated until the number of partitions is equal to the number of patterns in L. From the process, we obtain the new learning set L'. Another ANN **P** is trained on L'. The underlying theory of this process can be found in Wolpert[3,7]

When presented with a novel input pattern, **F** provides a solution u. Presenting **P** with the input pattern and a vector from the input to its nearest neighbor in the L gives an error bound ε such that the resultant solution is $u \pm \varepsilon$.

Results

For each training data set shown in Figures 2 and 3, **F** is trained and then provides outputs for untrained inputs. Corresponding to the outputs, error bounds are predicted by **P**. The predicted error bounds can be compared to the real RMS errors representing the true errors of **F**'s outputs. Figures 2 and 3, respectively, shows the results of a nonuniformly-selected data set and a uniformly-selected data set. Clearly, Figure 3 provides better results. It is not always possible, however, to obtain such well-behaved training data. These results are typical and the methodology can be applied to any ANN model[6].

References

1. E. B. Bartlett, R. Uhrig, "Nuclear Power Plant Status Diagnostics Using an Artificial Neural Networks," Nuclear Technology, 97, 272, (1992)

2. E. B. Bartlett et al, "Neural Network Recognition of Nuclear Power Plant Transients," Annual Report to Department of Energy, DOE/ER/75700-1 (1993)

3. D. H. Wolpert, "Stacked Generalization," Neural Networks, Vol. 5, p.241-259 (1992)

4. K. Kim, T. A. Aljundi, E. B. Bartlett, "Nuclear Power Plant Fault-Diagnostic Using Artificial Neural Networks," Intelligent Engineering Systems through Artificial Neural Networks, C. I. Dagli et al, editors, ASME Press, N.Y., 751-756 (1992)

5. K. Kim, T. A. Aljundi, E. B. Bartlett, "Confirmation of Artificial Neural Networks: Nuclear Power Plant Fault Diagnostics," ANS Transaction, 66, p.112-114 (1992)

6. K. Kim, E. B. Bartlett, "Error Prediction for a Nuclear Power Plant Fault-Diagnostic Advisor using Neural Networks." Submitted to Nuclear Technology for publication.

7. D. H. WOLPERT, "A Mathematical Theory of Generalization: Part I and Part II," Complex Systems, 4, p.151-249 (1990)

FIGURE CAPTIONS

Figure 1.

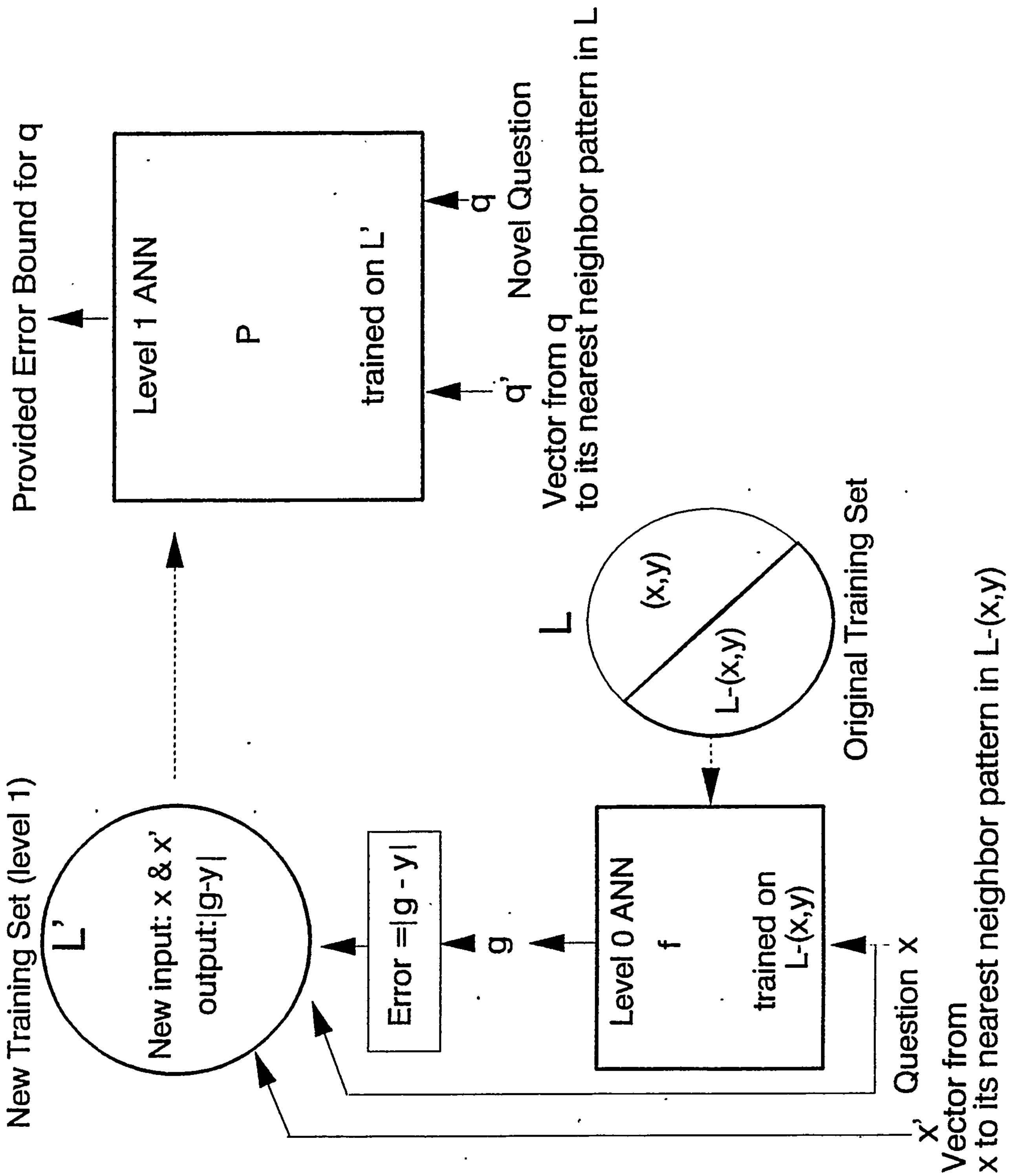
Illustration of the stacked generalization method to provide error bounds on novel solutions obtained from ANNs. Note that the nearest pattern is chosen in L-(x,y) such that the pattern has the minimum of Euclidean distance from the x.

Figure 2.

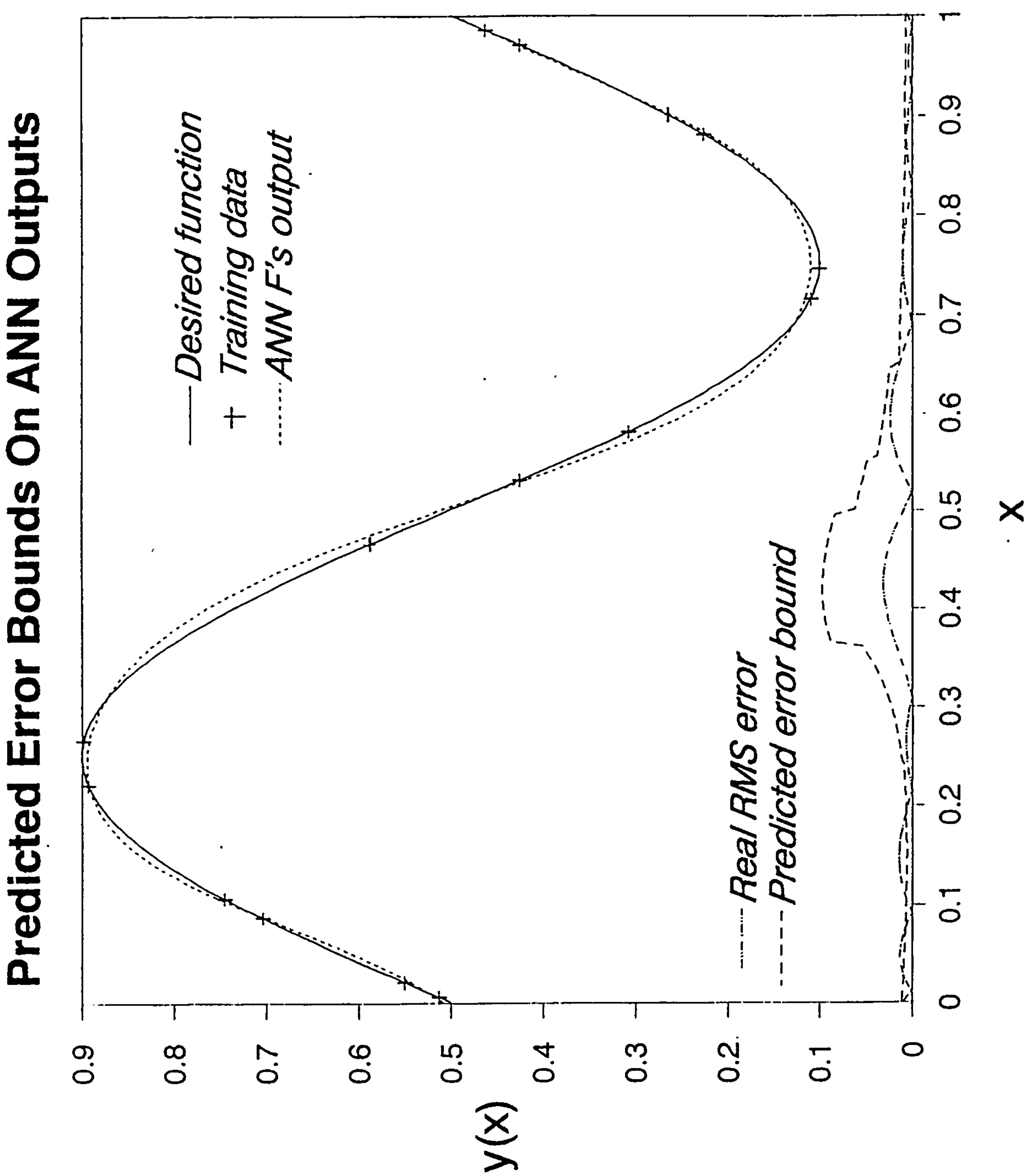
Predicted error bounds on an ANN's output. 15 data points of a sin function $y(x) = 0.4\sin(\pi x)+0.5$ are used to train the two-hidden-layer ANN, **F** (1-6-4-1). The single-hidden-layer ANN, **P** (2-5-1), trained on the new training set predicts the error bounds on **F**'s solutions. The real RMS error is based on the difference between **F**'s output and the desired function y(x), which represents the true error of the ANN for the untrained data points. The true error is compared to the predicted error bound obtained in this work.

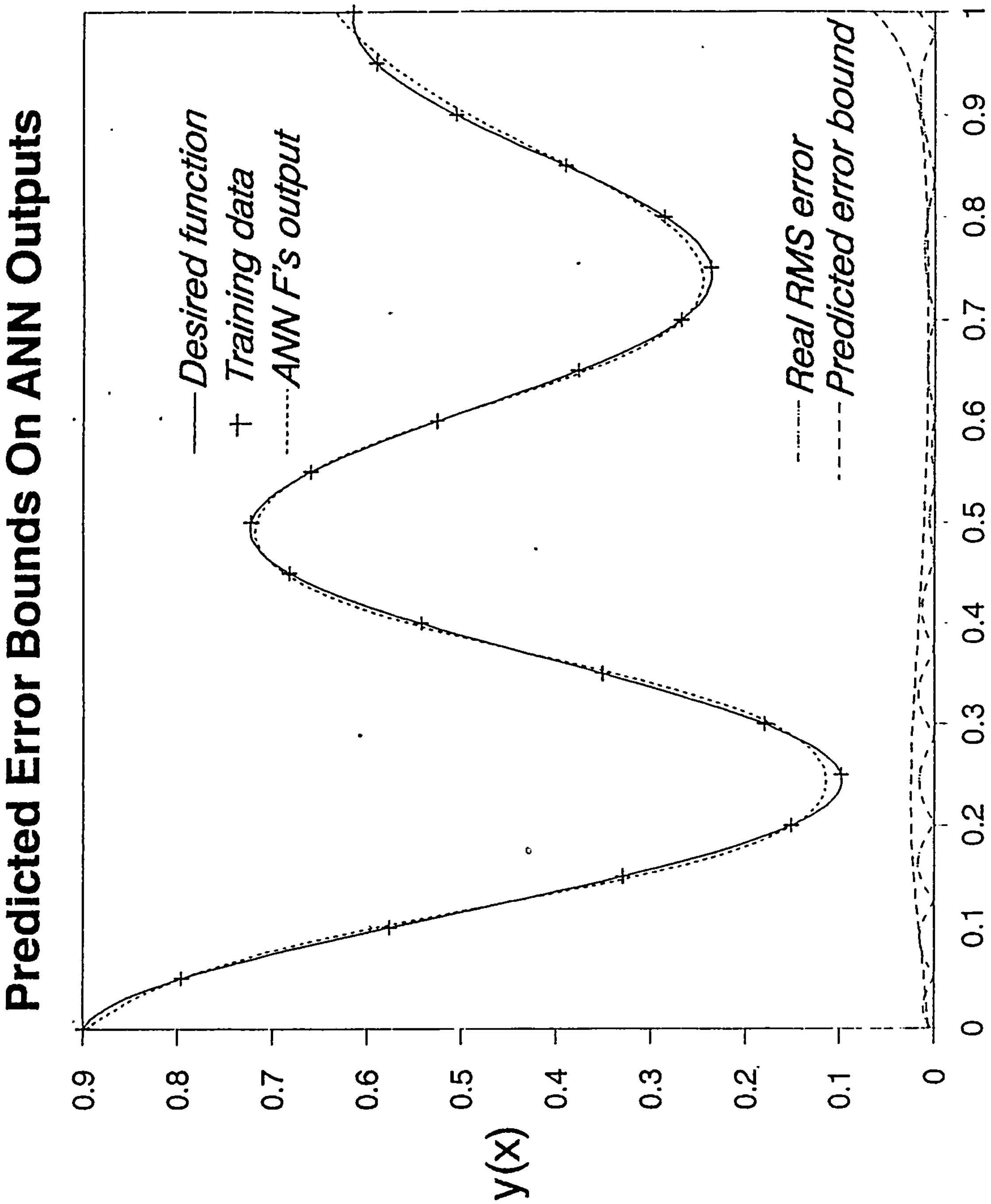
Figure 3.

Predicted error bounds on an ANN's output. 21 data points of a damped cosine function $y(x) = 0.45[1+\cos(4\pi x)\exp(-x)]$ are used to train the ANN, **F** (1-7-5-1). The ANN, **P** (2-5-1), trained on the new training set provides the error bounds on the **F**'s solutions.



Provided Error Bound for q
Level 1 ANN
P
trained on L'
q'
q
Novel Question
Vector from q
to its nearest neighbor pattern in L
New Training Set (level 1)
L'
New input: x & x'
output:|g-y|
Error =|g - y|
g
Level 0 ANN
f
trained on
L-(x,y)
L
(x,y)
L-(x,y)
Original Training Set
Question x
x'
Vector from
x to its nearest neighbor pattern in L-(x,y)


Predicted Error Bounds On ANN Outputs
Desired function
Training data
ANN F's output
Real RMS error
Predicted error bound
y(x)
X
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



Predicted Error Bounds On ANN Outputs
Desired function
Training data
ANN F's output
Real RMS error
Predicted error bound
y(x)
x
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1