

**SHORT TERM STREAMFLOW FORECASTING  
USING ARTIFICIAL NEURAL NETWORKS**

by

**Cameron M. Zealand**

A Thesis  
Submitted to the Faculty of Graduate Studies  
in Partial Fulfillment of the Requirements  
for the Degree of

**MASTER OF SCIENCE  
IN  
CIVIL ENGINEERING**

**Department of Civil and Geological Engineering  
University of Manitoba  
Winnipeg, Manitoba**

© July, 1997



National Library  
of Canada

Acquisitions and  
Bibliographic Services

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

Bibliothèque nationale  
du Canada

Acquisitions et  
services bibliographiques

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file Votre référence*

*Our file Notre référence*

**The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.**

**The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.**

**L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.**

**L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.**

0-612-23568-8

**THE UNIVERSITY OF MANITOBA  
FACULTY OF GRADUATE STUDIES  
\*\*\*\*\*  
COPYRIGHT PERMISSION PAGE**

**SHORT TERM STREAMFLOW FORECASTING USING  
ARTIFICIAL NEURAL NETWORKS**

**by**

**CAMERON M. ZEALAND**

**A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University  
of Manitoba in partial fulfillment of the requirements of the degree  
MASTER of SCIENCE**

**CAMERON M. ZEALAND 1997 (c)**

**Permission has been granted to the Library of The University of Manitoba to lend or sell  
copies of this thesis/practicum, to the National Library of Canada to microfilm this thesis  
and to lend or sell copies of the film, and to Dissertations Abstracts International to publish  
an abstract of this thesis/practicum.**

**The author reserves other publication rights, and neither this thesis/practicum nor  
extensive extracts from it may be printed or otherwise reproduced without the author's  
written permission.**

## **ABSTRACT**

Many of the activities associated with the planning and operation of water resource systems require forecasts of future events. For the hydrologic component that forms the input for water resource systems, there is a need for both short term and long term forecasts of streamflow events in order to optimize the real-time operation of the system or to plan for future expansion.

The main objective of this research is to investigate the utility of Artificial Neural Networks (ANNs) for short term forecasting of streamflow. Short term is defined as weekly time steps up to a time horizon of one month ahead. The work explores the capabilities of ANNs and compares the performance of this tool to conventional approaches used to forecast streamflow events one, two, three and four weeks in advance. A number of issues associated with the configuration of the ANN are examined to determine the preferred approach for implementing this technology in the forecasting mode. The performance of the ANN for the forecasting task is evaluated for a range of streamflow conditions in order to test the capabilities of ANNs in a realistic setting. The capabilities of the ANN model are compared to those of more traditional forecasting methods to ascertain the relative merits of each approach.

The specific issues addressed in this research include the applicability of ANNs for forecasting hydrologic variables; the approaches that are best used for identifying the appropriate structure for the ANN (the number of layers and the number of nodes), the preferred strategy for training the ANN; and the transferability of the ANN technology to another subwatershed. The application of the ANN approach is to a portion of the Winnipeg River system in Northwestern Ontario, Canada. This system is large in spatial extent and has a hydrometric data collection network that is very sparse. In this type of setting, it is often difficult to obtain reliable forecasts of future streamflow events due, in part, to the lack of accurate data for the required model inputs. As well, the remote location and complex hydraulic relationships of many of the sites contribute to a poor

quality streamflow gauging record. ANNs have been found to be effective in situations with noisy data. A perceived strength of ANNs is the capability for representing complex, nonlinear relationships as well as being able to model interaction effects. This capability is expected to be beneficial for forecasting since the relationship between the input variables and the resulting output (streamflow) is typically quite complex.

The results obtained were most promising. A very close fit was obtained during the training phase and the ANNs developed consistently outperformed the Winnipeg Flow Forecasting System (WIFFS) model during the testing phase in all of the four forecast lead-times for the Namakan Lake subwatershed located in the Winnipeg River Basin in Northwestern Ontario, Canada. The improvement in the Root-Mean-Squared Errors for the eight years of test data varied from 5 cms in the four-week lead forecasts to 12.1 cms in the two-week lead forecasts. Results also indicate that there was a 10 cms improvement in the Root-Mean-Squared-Error for the eight years of test data for the Rainy Lake subwatershed located in the Winnipeg River Basin in Northwestern Ontario, Canada (from the WIFFS model to the ANN model) for the one-week lead. This indicated that the ANN technology developed in this research is transferable to another subwatershed.

## **ACKNOWLEDGMENTS**

Throughout the course of my Master's degree studies, many people provided me with a great deal of support in many different ways. At this time I would like to express my deepest gratitude to each and every one of them. It is impossible to list the names of everyone and the means in which they aided me. However, I would like to specifically thank a select number of those people who assisted me throughout the course of my studies.

I would like to thank my entire family for their never-ending moral support and encouragement throughout the pursuit of my academic endeavors. A special thanks to my parents who have shown me the rewards of hard work and have always provided me with inspiration throughout my studies as well as my two sisters for their unconditional love.

To my advisors Dr. D. H. Burn and Dr. S.P. Simonovic, I am indebted for accepting me as a graduate student, advising me in all of my academic pursuits, and for the continuous support throughout my studies.

I would like to thank my friends and colleagues for their assistance, encouragement and moral support throughout the last two years. I would especially like to thank my friends and colleagues at the FIDS (Facility for Intelligent Decision Support in Water Resources) for making the time I spent at the University of Manitoba enjoyable and very memorable.

I would also like to thank Dave Cormie, Karl Reznichuk, John Crawford and the rest of the people at Manitoba Hydro who provided financial support and guidance for this thesis project. Their interest in the research related to my thesis, and confidence in my ability to successfully fulfill their expectations was a major factor contributing to the commencement and completion of this project

Finally, I would like to thank NSERC and the NSERC committee for selecting me as a Post-Graduate Scholarship recipient towards pursuit of my Master's degree.

# TABLE OF CONTENTS

	<b>Page</b>
<b>ABSTRACT .....</b>	<b>ii</b>
<b>ACKNOWLEDGMENTS.....</b>	<b>iv</b>
<b>TABLE OF CONTENTS.....</b>	<b>v</b>
<b>LIST OF FIGURES.....</b>	<b>x</b>
<b>LIST OF TABLES .....</b>	<b>xii</b>
<b>CHAPTER 1 .....</b>	<b>1</b>
<b>1 INTRODUCTION .....</b>	<b>1</b>
<b>1.1 PROBLEM STATEMENT .....</b>	<b>1</b>
<b>1.2 PURPOSE OF RESEARCH.....</b>	<b>2</b>
<b>1.3 SCOPE OF THE INVESTGATION .....</b>	<b>2</b>
<b>CHAPTER 2 .....</b>	<b>6</b>
<b>2 LITERATURE REVIEW .....</b>	<b>6</b>
<b>2.1 PHYSICALLY BASED MODELS.....</b>	<b>6</b>
<b>2.2 TIME-SERIES MODELS.....</b>	<b>8</b>
<b>2.2.1 General.....</b>	<b>8</b>
<b>2.2.2 Time-Series Modelling Procedures.....</b>	<b>8</b>
<b>2.2.3 Stochastic/Time-Series Models.....</b>	<b>9</b>

2.3	ARTIFICIAL NEURAL NETWORK MODELS.....	10
<b>CHAPTER 3</b>	.....	<b>13</b>
<b>3</b>	<b>BACKGROUND ON ARTIFICIAL NEURAL NETWORKS</b> .....	<b>13</b>
3.1	GENERAL.....	13
3.2	ANN ARCHITECTURE .....	17
3.2.1	Multilayer Feed Forward ANN Architecture .....	18
3.2.2	The General Processing Element.....	21
3.2.3	Transfer Functions .....	22
3.3	NETWORK CONFIGURATION.....	24
3.4	NETWORK TRAINING.....	25
3.4.1	Backpropagation Training Algorithm .....	27
3.4.2	Generalized Delta Rule.....	27
3.4.3	Gradient Descent Method .....	28
3.5	NETWORK TESTING .....	31
3.6	ANN MODEL DEVELOPMENT .....	33
3.7	BRAINCEL (ANN TOOL).....	34
3.7.1	Auto Expert User Mode.....	35
3.7.2	Automated Best Net Search .....	35
3.7.3	Professional User Mode .....	37
<b>CHAPTER 4</b>	.....	<b>38</b>
<b>4</b>	<b>CASE STUDY APPLICATION</b> .....	<b>38</b>
4.1	GENERAL.....	38
4.2	DESCRIPTION OF TEST LOCATION.....	38



<b>4.3</b>	<b>WINNIPEG FLOW FORECASTING SYSTEM (WIFFS)</b>	<b>42</b>
4.3.1	Model Type	42
4.3.2	Model Inputs	44
4.3.3	Available Data and Resources	45
<b>4.4</b>	<b>TYPES OF MODEL COMPARISONS</b>	<b>46</b>
4.4.1	Experiment #1	46
4.4.2	Experiment #2	46
<b>4.5</b>	<b>TIME-SERIES FLOW FORECASTING USING ANNs</b>	<b>47</b>
4.5.1	Single Predictor	47
4.5.2	Multiple Predictors Series	48
<b>4.6</b>	<b>ANN MODEL IDENTIFICATION</b>	<b>50</b>
4.6.1	Experiment #1	50
4.6.2	Experiment #2	54
<b>CHAPTER 5</b>		<b>58</b>
<b>5</b>	<b>RESULTS AND ANALYSIS</b>	<b>58</b>
<b>5.1</b>	<b>RESULTS OF FLOW FORECASTING</b>	<b>58</b>
5.1.1	Experiment #1	58
5.1.2	Experiment #2	66
<b>5.2</b>	<b>COMPARISON OF RESULTS</b>	<b>73</b>
5.2.1	Experiment #1	75
5.2.2	Experiment #2	76
5.2.2.1	<i>R.M.S.E</i>	76
5.2.2.2	<i>Forecasted Total Yearly Volume</i>	77
5.2.2.3	<i>Forecasted Peak Flow Magnitude</i>	78
5.2.2.4	<i>Forecasted Location of Peak Flow</i>	79
<b>5.3</b>	<b>SEPARATION OF DATA INTO TRAIN/TEST SETS</b>	<b>80</b>

<b>5.4</b>	<b>SUMMARY .....</b>	<b>85</b>
<b>CHAPTER 6 .....</b>	<b>86</b>	
<b>6</b>	<b>TRANSFERABILITY OF THE RESULTS .....</b>	<b>86</b>
<b>6.1</b>	<b>ANN MODEL IDENTIFICATION.....</b>	<b>86</b>
<b>6.2</b>	<b>RESULTS OF FLOW FORECASTING.....</b>	<b>86</b>
<b>6.3</b>	<b>COMPARATIVE ANALYSIS OF WIFFS AND ANN MODELS .....</b>	<b>89</b>
<b>6.3.1</b>	<b>R.M.S.E.....</b>	<b>89</b>
<b>6.3.2</b>	<b>Forecasted Total Yearly Volume.....</b>	<b>90</b>
<b>6.3.3</b>	<b>Forecasted Peak Flow Magnitude.....</b>	<b>91</b>
<b>6.3.4</b>	<b>Forecasted Location of Peak Flow .....</b>	<b>92</b>
<b>6.4</b>	<b>SUMMARY .....</b>	<b>93</b>
<b>CHAPTER 7 .....</b>	<b>95</b>	
<b>7</b>	<b>CONCLUSIONS AND RECOMMENDATIONS .....</b>	<b>95</b>
<b>CHAPTER 8 .....</b>	<b>98</b>	
<b>8</b>	<b>FUTURE WORK.....</b>	<b>98</b>
<b>REFERENCES .....</b>	<b>99</b>	
<b>APPENDIX A: NAMAKAN LAKE HISTORICAL DATA.....</b>	<b>104</b>	

<b>APPENDIX B: NAMAKAN LAKE TRAINING RESULTS, EXPERIMENT #1 .....</b>	<b>111</b>
<b>APPENDIX C: NAMAKAN LAKE TRAINING RESULTS, EXPERIMENT #2 .....</b>	<b>128</b>
<b>APPENDIX D: NAMAKAN LAKE SCATTER PLOTS FOR TRAINING SETS.....</b>	<b>145</b>
<b>APPENDIX E: RAINY LAKE LOCAL HISTORICAL DATA....</b>	<b>150</b>
<b>APPENDIX F: RAINY LAKE LOCAL TRAINING RESULTS...</b>	<b>157</b>

## LIST OF FIGURES

Page

---

Figure 3.1. An example of a complex pattern (Freeman & Skapura, 1991) .....	14
Figure 3.2. A Vector Mapper .....	18
Figure 3.3. Multilayer Feed-Forward ANN and Processing Element Architectures .....	19
Figure 3.4. Sigmoidal Logistic Function (Freeman & Skapura, 1991) .....	23
Figure 3.5. Bowl-Shaped Error Surface (Hammerstrom, 1993) .....	29
Figure 3.6. A cross-section of a hypothetical error surface in weight space (Freeman & Skapura, 1991) .....	29
Figure 3.7. Training-set/Test-set error (Hammerstrom, 1993) .....	32
Figure 4.1. Winnipeg River Basin (LWCB brochure, 1994) .....	39
Figure 4.2. Namakan Lake Subwatershed (LWCB brochure, 1994) .....	41
Figure 4.3. Components of WIFFS Model .....	43
Figure 4.4. Univariate time-series prediction model .....	48
Figure 4.5. Multivariate time-series prediction model .....	49
Figure 4.6. Experiment #1, Variation of MSE for (1) training set and (2) testing set, with the number of hidden nodes for the one-week ahead forecast. ....	54
Figure 4.7. Experiment #2.2, Variation of MSE for (1) training set and (2) testing set, with the number of inputs - for the one-week ahead forecast. ....	56
Figure 5.1. Best 1-week forecast of local inflow into Namakan Lake for 1960-64 & 1986-88 using the Exp.#1 ANN model .....	60
Figure 5.2. Best 2-week forecast of local inflow into Namakan Lake for 1960-64 & 1986-88 using the Exp.#1 ANN model .....	61
Figure 5.3. Best 3-week forecast of local inflow into Namakan Lake for 1960-64 & 1986-88 using the Exp.#1 ANN model .....	62
Figure 5.4. Best 4-week forecast of local inflow into Namakan Lake for 1960-64 & 1986-88 using the Exp.#1 ANN model .....	63

Figure 5.5. Observed versus Best 1-week Forecast of local inflow to Namakan Lake for 1960-64 & 1986-88 using Exp. #1 ANN.....	64
Figure 5.6. Observed versus Best 2-week Forecast of local inflow to Namakan Lake for 1960-64 & 1986-88 using Exp. #1 ANN.....	65
Figure 5.7. Observed versus Best 3-week Forecast of local inflow to Namakan Lake for 1960-64 & 1986-88 using Exp. #1 ANN.....	65
Figure 5.8. Observed versus Best 4-week Forecast of local inflow to Namakan Lake for 1960-64 & 1986-88 using Exp. #1 ANN.....	66
Figure 5.9. Best 1-week forecast of local inflow into Namakan Lake for 1960-64 & 1986-88 using the Exp. #2 ANN model.....	67
Figure 5.10. Best 2-week forecast of local inflow into Namakan Lake for 1960-64 & 1986-88 using the Exp. #2 ANN model.....	68
Figure 5.11. Best 3-week forecast of local inflow into Namakan Lake for 1960-64 & 1986-88 using the Exp. #2 ANN model.....	69
Figure 5.12. Best 4-week forecast of local inflow into Namakan Lake for 1960-64 & 1986-88 using the Exp. #2 ANN model.....	70
Figure 5.13. Observed versus Best 1-week Forecast of local inflow to Namakan Lake for 1960-64 & 1986-88 using Exp. #2 ANN.....	71
Figure 5.14. Observed versus Best 2-week Forecast of local inflow to Namakan Lake for 1960-64 & 1986-88 using Exp. #2 ANN.....	71
Figure 5.15. Observed versus Best 3-week Forecast of local inflow to Namakan Lake for 1960-64 & 1986-88 using Exp. #2 ANN.....	72
Figure 5.16. Observed versus Best 4-week Forecast of local inflow to Namakan Lake for 1960-64 & 1986-88 using Exp. #2 ANN.....	72
Figure 6.1. Best 1-week forecast of local inflow into Rainy Lake for 1960-64 & 1986-88 using the ANN model.....	87
Figure 6.2. Observed versus Best 1-week Forecast of local inflow to Rainy Lake for 1960-64 & 1986-88 (test set) using the ANN model.....	88
Figure 6.3. Observed versus Best 1-week Forecast of local inflow to Rainy Lake for 1965-85 (train set) using the ANN model.....	88

## **LIST OF TABLES**

	<b>Page</b>
<hr style="border: 1px solid black;"/>	
Table 5.1. Root Mean Squared Forecast Error, Experiment #1.....	75
Table 5.2. Root Mean Squared Forecast Error, Experiment #2.....	76
Table 5.3. Percent of Total Yearly Volume Forecasted, Experiment #2.....	77
Table 5.4. Percent Error in Forecast of Peak Flow Magnitude, Experiment #2.....	78
Table 5.5. Error in Location of Forecasted Peak Flow, Experiment #2.....	80
Table 5.6. Breakdown of Training/Testing Data Set.....	81
Table 6.1. Root Mean Squared Forecast Error, Rainy Lake.....	90
Table 6.2. Percent of Total Yearly Volume Forecasted, Rainy Lake.....	91
Table 6.3. Percent Error in Forecast of Peak Flow Magnitude, Rainy Lake.....	92
Table 6.4. Error in Location of Forecasted Peak Flow, Rainy Lake.....	92

## CHAPTER 1

### 1 INTRODUCTION

#### 1.1 PROBLEM STATEMENT

Many of the activities associated with the planning and operation of a water resource system require forecasts of future events. Examples include forecasts of future demand for electricity, forecasts of the selling price associated with short or long term power sales, and forecasts of costs associated with alternative energy sources, to name but a few. For the hydrologic component that forms the input for hydroelectric power generation, there is a need for both short term and long term forecasts of streamflow events in order to optimize the system or to plan for future generation expansion. The Nelson-Churchill drainage basin is the hydraulic system that is used by Manitoba Hydro for power generation. This system is large in spatial extent (i.e. the locations of data collection are few and remote from the forecaster) and has a hydrometric data collection network that is very sparse. These conditions can result in considerable uncertainty in the data which are available. In this type of setting, it is often difficult to obtain reliable forecasts of future streamflow events due to the lack of accurate data for the required model inputs. As well, the remote location and complex hydraulic relationships of many of the sites contribute to a poor quality streamflow gauging record. As such, it is important that alternative approaches to estimating future streamflow events be explored in this area.

Many of the techniques currently used in modelling hydrological time-series and generating synthetic streamflows assume a linear relationship among variables. *Chakraborty* [1992] states that, in the real world, temporal variations in data do not exhibit simple regularities and are difficult to analyze and predict accurately. Linear relationships for describing the behavior of such complex data are often found to be inadequate. Therefore, it seems necessary to use nonlinear models such as Artificial

Neural Networks (ANNs), which are suited to complex nonlinear problems, for the analysis of real world temporal data.

## **1.2 PURPOSE OF RESEARCH**

The main objective of this research is to explore the use of ANNs for short term streamflow forecasting. Comparisons will be made between the performance of different neural network configurations and a model based on a more traditional forecasting approach. Conventional models for streamflow forecasting typically involve a number of physical variables that function as inputs. A physical variable that is not very useful for forecasting on its own can often be useful when used in conjunction with other variables. Given the number of physical variables that could be considered as potentially relevant, it is apparent that a very large number of different combinations of both variables and mathematical relationships that link them together are available when developing a streamflow forecasting model. Determining an appropriate model structure by a trial-and-error process is therefore not always practical. In this context, the power of ANNs arises from the capability for constructing complicated indicators (nonlinear models) for multivariate time-series without resorting to the use of complex statistics, system identification theory, or other advanced mathematical techniques.

## **1.3 SCOPE OF THE INVESTGATION**

The main focus of this research is concerned with the development of ANN models for short term streamflow forecasting and the development of general methodologies for using these models for any catchment area. This work will involve configuring, training, and testing an ANN for a case study area. The relative performance characteristics for different ANNs will be assessed to ascertain the preferred alternative for the forecasting application. In addition, the performance of the preferred ANN will be compared to the performance capabilities of a more traditional forecasting approach for a representative range of hydrologic conditions.



The configuration and training of an ANN is analogous to the formulation and calibration of a model of the type traditionally used in hydrologic applications (e.g. a rainfall-runoff model). The configuring of the ANN involves determining the most appropriate structure for the network (i.e. identifying the number of layers, the number of nodes in each layer and the type of non-linear transformation between layers). The training of the ANN involves feeding the network a variety of input values and comparing the predicted output from the network to the actual (observed) output. The network will then alter the weights that it applies to each input node and hidden node of the network in order to more closely match the observed output. There are a number of currently unresolved issues associated with configuring and training an ANN. *Karunanithi et al.* [1994] state that there has been comparatively little work done in determining a systematic approach for identifying the appropriate number of layers and nodes in an ANN. In addition, it has often been observed that the selection of training sets to submit to a neural network and the order in which they are presented can affect the final results of the training process and hence the overall caliber of the performance of the ANN. It is the intent of this research to address these issues in a systematic manner.

Testing of an ANN is analogous to the verification of a model of the type traditionally used in hydrologic applications (e.g. a rainfall-runoff model). Verification of an ANN involves submitting an independent set of input data (i.e. data that was not used in the training process) to the neural network and evaluating the prediction capability of the network for this set of data. The level of agreement between the forecasted (predicted) output of the network and the actual (observed) output is used as an indicator of the performance of the ANN. Unsatisfactory performance would indicate that the ANN structure should be altered and the training exercise repeated. It is important at this stage to ensure that the verification data set contains a sufficient range of input conditions. This is to ensure that the performance of the model is being properly evaluated over the full range of conditions that might be encountered in an operational mode. The ultimate evaluation of the ANN model will come through a comparison of the network model with a traditional forecasting tool. The two modelling approaches will be compared not only in

terms of the accuracy of the forecasting results but also in terms of the ease of use of the two approaches.

There are several issues that will be addressed in the proposed application of ANNs to short term streamflow forecasting. The nature and the number of the model inputs must be determined. Inputs to short term streamflow forecasting models typically include observations of streamflow for the present and past time intervals as well as variables representing present and past meteorological inputs (e.g. temperature, precipitation and contributions to snow melt). A perceived strength of ANNs is the capability to model complex nonlinear relationships as well as interaction effects among input variables. This capability would be expected to be beneficial for forecasting since the relationship between input variables and the resulting output (streamflow) is typically quite complex. A potential disadvantage of an ANN type model is that it is often difficult to directly interpret the model parameters (connection weights) from a physical basis (i.e. the model is structured like a black box). This can be problematic if the training data set is not chosen properly such that the model becomes very efficient at mimicking the training data set but has a very poor generalization capability. It will therefore be important to carefully select the data sets included in the training data set and to ensure that a proper balance is maintained between reproduction and generalization capabilities.

Contained in this thesis is a description of the research performed towards the configuration, training, testing, and evaluation of ANN model alternatives for the case study area. Chapter 2 begins with an overview of traditional streamflow forecasting techniques such as conventional physical modelling and ARMA type modelling as well as what has been done in the application of ANNs in solving water resource problems. Chapter 3 describes essential background information on ANNs. This includes a description of the structure and terminology of an ANN. The application of the forecast technique to the Namakan Lake subwatershed, located in Northwestern Ontario, Canada, is described in Chapter 4. A brief description of the case study area is presented, followed by a discussion of the systematic process invoked to obtain the “optimal” ANN structure.

Chapter 5 describes the method of evaluation and goes on to present the analysis of the results obtained from the ANNs developed in chapter 4. Some discussion is given to the topic of how to separate the data set into the training and testing ranges to guarantee satisfactory network performance in the operational mode. Chapter 6 investigates the transferability of the systematic approach for configuring an ANN, developed in Chapter 4, to the Rainy Lake Local subwatershed, located in Northwestern Ontario, Canada. Recommendations are given on the procedures to be followed in applying the technique to similar forecasting applications. Chapter 7 presents conclusions based on the analysis of the case study and summarizes some of the experience gained in applying the technique to a real-world application. Finally, Chapter 8 provides the reader with recommendations for future research directions.

## CHAPTER 2

### 2 LITERATURE REVIEW

Generally, forecasting models can be divided into statistical and physically based approaches. Statistical approaches determine relationships between historical data sets, whereas physically based approaches model the underlying physical processes directly. Back-propagation networks are closely related to statistical models and according to *Maier and Dandy* [1996] are the type of ANNs most suited to forecasting applications. The following sections present a brief discussion on the various types of forecasting models with an emphasis primarily on streamflow forecasting. Different types of conventional and time-series models are discussed as well as their applicability to various types of problems. Finally, a discussion of the most recent technology that is gaining importance and recognition in the field of streamflow forecasting is presented.

#### 2.1 PHYSICALLY BASED MODELS

A *conceptual* model is specifically designed to mathematically simulate the general internal sub-processes and *physical* mechanisms that govern the hydrological cycle. These types of models usually incorporate simplified forms of physical laws and are generally nonlinear, time-invariant, and deterministic, with parameters that are representative of watershed characteristics [*Hsu*, 1995] but ignore the spatially distributed, time-varying, and stochastic properties of the rainfall-runoff (R-R) process. These conventional streamflow models are based on the physical descriptions of the hydrologic and hydraulic processes that govern the rainfall-runoff process. They are very powerful attempts to represent, in a simplified manner, the known physical process occurring in the rainfall-runoff transformation and are commonly referred to as conceptual rainfall-runoff models. *Kitanidis and Bras* [1980 a, b] state that conceptual watershed models are generally reported to be reliable in forecasting the most important features of the hydrograph, namely, the beginning of the rising limb, the time and height of the peak, and the volume

of flow. However, the implementation and calibration of such a model can typically present various difficulties [Duan *et al.*, 1992], requiring sophisticated mathematical tools [Duan *et al.*, 1992 & 1994; Sorooshian *et al.*, 1993], significant amounts of calibration data [Yapo *et al.*, 1995], and some degree of expertise and experience with the model [Hsu *et al.*, 1995]. The first type of physically based models is the hydrometric data-based model involving only streamflows. These methods involve correlating stages or discharges and volumes, input/output models such as unit hydrograph approaches, and hydraulic routing using dynamic wave, diffusion wave, kinematic wave, Muskingum method, impulse response function, or storage models. The second, more common type of conceptual model, is the hydrometeorologic and hydrometric data-based model involving precipitation-to-runoff and streamflow processes. These complex rainfall-runoff models represent the various water storage terms (interception, evapotranspiration, snowmelt, interflow, groundwater baseflow, and surface runoff from rainfall and snowmelt) in varying levels of complexity as well as indices such as Antecedent Precipitation Index (API). The most complex of the rainfall-runoff models currently being used are the conceptual storage models, a representative example being, the U.S. National Weather Service River Forecast System. Singh [1992] provides a list of conceptual models that attempt to model the underlying process of a system. The application of the underlying processes are based on the discovery of strong empirical regularities gained through observations of the system. These types of models provide a very powerful and accurate means of prediction, but discovering the processes underlying the behavior of a system is often a difficult task. The problem with conceptual models is that empirical regularities or periodicities are not always evident and can often be masked by noise. Tong *et al.* [1985] state that some of the problems with a conceptual model are that it is often very complex and possesses a large number of parameters related to physical phenomena, all of which must be calibrated. Further, due to the great complexity of natural systems, the conceptual model is only a crude approximation to reality.

## **2.2 TIME-SERIES MODELS**

### **2.2.1 General**

Conceptual models have proven their importance in the understanding of hydrologic processes, but there are many practical situations such as streamflow forecasting where the main concern is with making accurate predictions at specific watershed locations. In order to understand and model how one or more inputs to a given system control various outputs, engineers and scientists take measurements over time. For a given input or output variable that is being monitored, a set of observations arranged chronologically is called a *time-series*. In time-series analysis, stochastic or time-series models are fitted to one or more of the time-series describing the system for purposes which include forecasting, generating synthetic sequences for use in simulation studies, and investigating and modelling the underlying characteristics of the system under study. In fact, time-series and statistical models constitute one of the most frequently employed set of tools used in water resources planning and management [Hipel, 1985].

### **2.2.2 Time-Series Modelling Procedures**

Multivariate time-series modelling is comprised of three basic stages, namely: model identification, parameter estimation and diagnostic checking for model accuracy. This procedure, formalized by [Box and Jenkins, 1970] and advocated by [Salas et al., 1980], is an iterative procedure of model building to ensure satisfactory model fitting and utilization. The model identification stage is divided into model structure specification and model order determination. Model structure specification is geared to finding the structural form of the model or type of dependence relationship to be incorporated in the model. Model order determination is concerned with identifying the number of lags and number of moving average parameters. Some of the tools reviewed in the literature include cross-correlation analysis, partial-autocorrelation analysis and likelihood ratio test. After identifying possible models, the parameter estimation stage follows. The methods of moments and maximum likelihood are among the most popular methods of parameter estimation that have been suggested in the literature. Once the model parameters have been estimated, diagnostic checks are necessary in order to see whether the selected model

is appropriate, whether it is better than other competing models, and whether to search for model improvements. Some common diagnostic checks include model implementation as well as testing the robustness of the model. For instance, the model may be implemented according to its intended utilization, such as data augmentation, generation, or forecasting, and examine how well the model performs. Robustness tests may be applied to see if the model preserves properties not explicitly parameterized in the model (i.e. flood/drought characteristics).

### **2.2.3 Stochastic/Time-Series Models**

Stochastic forecasting models can be broken up into five different general types. They are Autoregressive (AR) models, Moving average (MA) models, Autoregressive moving average (ARMA) models, Nonlinear (threshold) time-series models; and Adaptive algorithm models [Szollosi-Nagy, 1987]. Univariate and multivariate time-series analysis is an important statistical tool to study the behavior of time dependent data and forecast future values depending on the history of variations in the data [Chakraborty *et al.*, 1992]. Most of the time-series modelling procedures fall within the framework of multivariate autoregressive moving average (ARMA) models [Raman and Sunilkumar, 1995]. Traditionally, the class of ARMA models have been the statistical method most widely used for modelling water resources time-series [Maier and Dandy, 1996] an example of which is found in Bender *et al.*, 1992. In streamflow forecasting, time-series models are used to describe the stochastic structure of the time sequence of streamflows and precipitation values measured over time, in discrete or continuous time units. Time-series models are more practical than conceptual models because one is not required to understand the internal structure of the physical processes that are taking place in the system being modelled. The forecaster needs only to determine the order of the time-series equation. Forecasts using time-series methods have the property that they approach the long-term mean as the forecast lead-time increases. The rate at which these forecasts approach the mean depends on how far the initial value (the most recent observed flow) is from the long term mean. The limitation of univariate time-series methods in streamflow forecasting, is that the only information they incorporate is that which is present in past

flows. Therefore, in regions that are dominated by snowmelt runoff, a method that makes direct use of snow water storage is usually preferable. *Chakraborty* [1992] states that a more accurate method is that of a multivariate time-series, which consists of sequences of values of several contemporaneous variables changing with time. By studying many related variables together, rather than by studying just one, a better understanding of the phenomenon being modelled is obtained. However, for situations where snow storage is not a factor, univariate time-series methods have the advantage that they are easily implemented, and the forecast error variance can be readily computed [*Box and Jenkins*, 1976]. Several stochastic time-series models have been proposed for modelling hydrological time-series and generating synthetic streamflows. These include Autoregressive Moving Average (ARMA) models [*Box and Jenkins*, 1970], disaggregation models [*Valencia and Schaake*, 1973], and models based on conceptual pattern recognition [*Pam and Unny*, 1980]. *Stedinger and Taylor* [1982] studied the performance of five different models for streamflow simulation. All of these types of models have their merits and also have been criticized. These models are relatively easy to develop and implement and have been found to provide satisfactory predictions in many water resources applications [*Salas et al.*, 1980]. Most real world problems, however, involve temporal variations in data that do not exhibit simple regularities. Many of the available techniques do not attempt to represent the nonlinear dynamics inherent in the transformation of rainfall to runoff and as a result, these problems tend to be difficult to analyze and predict accurately. *Tong* [1983] describes some of the drawbacks of linear modelling for univariate time-series analysis. These include, for example, their inability to explain sudden bursts of very large amplitudes at irregular time intervals, which is a common occurrence in streamflow forecasting. *Tong* [1990] also addresses some of the problems with linear models for multivariate time-series analysis.

### **2.3 ARTIFICIAL NEURAL NETWORK MODELS**

Artificial neural networks (ANNs) have been successfully applied in a number of diverse fields. This next section will only focus on water resource applications of the ANN technology. In order to fit an ARMA-type model to a time-series, the data have to be



stationary [*Hipel*, 1985] and have to follow a normal distribution [*Irvine and Eberhardt*, 1992]. If this is not the case, techniques such as differencing [*Box and Jenkins*, 1976] have to be used to induce stationarity and the Box-Cox transformation has to be applied to obtain normally distributed data. When developing ANN models, the statistical distribution of the data does not have to be known [*Burke*, 1991] and non-stationarities in the data, such as trends and seasonal variations, are accounted for by the internal structure of the ANNs [*Maier and Dandy*, 1996]. ANNs are suited to complex problems, where the relationship between the variables being modelled are not well understood. ANNs differ from the traditional approaches in synthetic hydrology in the sense that they belong to a class of *data-driven* approaches, as opposed to traditional *model-driven* approaches such as ARMA-type models. Data-driven approaches have the ability to determine which model inputs are critical, so that there is no need for a prior knowledge about relationships between variables. Model-driven approaches, on the other hand, require some understanding of the problem, as the model order has to be determined before the unknown model parameters can be estimated [*Maier and Dandy*, 1996]. An ANN is a flexible mathematical structure, capable of identifying complex nonlinear relationships between input and output data sets [*Hsu et al.*, 1995]. They are relatively insensitive to noisy data, unlike ARMA-type models, as they have the ability to determine the underlying relationship between model inputs and outputs, resulting in good generalization ability. ANN models have been found to be useful in solving problems where the characteristics of the processes were difficult to describe using physical equations (i.e. conceptual models). ANNs are defined as one of the approaches in the area of artificial intelligence which use a black-box approach where the user need not know much about the underlying process being modelled. The analysis depends on available data, with little rationalization about possible interactions between the data. ANNs are constructed to obtain a prediction of system response without attempting to reach understanding or to provide insight into the nature of the phenomena. Relationships between variables, models, laws and predictions are considered after building a machine whose behavior simulates the data being studied. An ANN is a nonlinear mathematical structure capable of representing arbitrarily complex nonlinear processes that relate the inputs and outputs of any system. The success with

which ANNs have been used to model dynamic systems in areas of science and engineering suggests that the ANN approach may prove to be an effective and efficient way to model the streamflow process in situations where explicit knowledge of the internal hydrologic sub-processes of the watershed is not required. Some studies in which ANNs have been applied to problems involving rainfall-runoff modelling and weather and river flow prediction have been reported in the literature. *Lorrai and Sechi* [1995] verified the possibility of utilizing ANNs to predict rainfall-runoff when only information about the variation of the basic input variables, namely rainfall and temperature, is available. *Cheng and Noguchi* [1996] obtained better results modelling the rainfall-runoff process with ANNs using previous rainfall, soil moisture deficits, and runoff values as model inputs, when compared with that from the Xinanjiang model (a rainfall-runoff model). *Smith and Eli* [1995] applied ANNs to convert remotely sensed, spatially distributed rainfall patterns into rainfall rates, and hence into runoff for a given river basin. *Hsu et al.* [1995] showed that a nonlinear ANN model provided a better representation of the rainfall-runoff relationship of the medium-sized Leaf River basin near Collins, Mississippi, than the linear ARMAX (autoregressive moving average with exogenous inputs) time-series approach or the conceptual SAC-SMA (Sacramento soil moisture accounting) model. *French et al.* [1992] demonstrated that an ANN is capable of forecasting the complex temporal and spatial distribution of rainfall generated by a rainfall simulation model. *Jayawardena and Fernando* [1996] applied the ANN approach to predict evaporation rates (using the vapor pressure difference, temperature, solar radiation, and wind speed) that were found to be in better agreement with actual observations than those obtained from an empirical equation. *Karunanithi et al.* [1994] used ANNs for river flow prediction and *Raman and Sunilkumar* [1995] for forecasting multivariate water resources time-series. ANNs have also been applied to areas such as deriving a general operating policy for reservoirs *Raman and Chandramouli* [1996], prediction of water quality parameters *Maier and Dandy* [1996] and real-time forecasting of water quality *Dandy and Maier* [1996].

## CHAPTER 3

### 3 BACKGROUND ON ARTIFICIAL NEURAL NETWORKS

#### 3.1 GENERAL

Computers have traditionally been used to perform large numbers of calculations very quickly with high numerical accuracy. Computers run on a series of sequential algorithms or commands provided by the user. Therefore, if the user gives the computer an explicit set of instructions that includes an error, the computer will execute the commands and produce the wrong answer, i.e. the computer does not realize that it has made an error. Why can't the same machines that perform a million calculations per second *learn* from experience, rather than forever repeating a sequential algorithm generated by a human programmer? *Freeman and Skapura* [1991] state that if the only tool we have is a sequential computer, then we will naturally try to cast every problem in terms of sequential algorithms. Many real-world applications are not suited to this approach, yet many spend a great deal of time and effort trying to develop sophisticated algorithms to solve these problems, perhaps even failing to find an acceptable solution.

One of the most common problems encountered when sequential computer systems are used to perform an inherently parallel task is that of visual pattern recognition. Complex patterns contain many elements that, alone, reveal little information about the total pattern, but together represent easily recognizable (by humans) objects. These are the typical types of pattern that have proven to be the most difficult for computers to recognize. An example of this type of pattern is shown in Figure 3.1.

If you just focus on the black areas in the picture, the picture is meaningless. Yet, if you allow yourself to encompass all the components of the picture, you can see a commonly recognizable object. Furthermore, once you see the image, it is easy to see it again whenever you look at the picture. The illustration in Figure 3.1 is of a dalmatian seen in

profile, facing left, with head lowered sniffing the ground and walking towards a tree. Why is it that you can see the dog quickly, yet a computer cannot perform this task even though the switching time of modern electronic computers are extremely faster than the switching time of the cells that comprise our neurobiological systems?



**Figure 3.1.** An example of a complex pattern (Freeman & Skapura, 1991)

Part of the answer to this question is that the architecture of the human brain is significantly different from that of a conventional computer (i.e., the massive parallelism and interconnectivity observed in the biological systems is what allows the brain to perform complex pattern recognition in a few hundred milliseconds). Artificial Neural Networks (ANNs) are made up of the best of both worlds. They have the rapid calculation speeds of a computer and the complex pattern recognition capabilities of a human. They act on data by detecting some kind of underlying organization or pattern. These networks can recognize spatial, temporal, or other relationships and, by bridging the

gap between individual examples and general relationships, can perform tasks such as function modelling, estimation, or approximation.

ANNs offer valuable characteristics unavailable together elsewhere. First, they *infer* solutions from data without prior knowledge of the regularities in the data; they extract the regularities empirically. This characteristic is useful because gathering data does not require explaining it. Second, these networks *learn* the similarities among patterns directly from instances or examples of them. ANNs can modify their behavior in response to the environment (i.e. shown a set of inputs with corresponding desirable outputs, they self-adjust to produce consistent responses). Third, ANNs can *generalize* from previous examples to new ones. That is, once the network has been trained, they can respond correctly to patterns that are only broadly similar to the original training patterns (i.e. its response can be insensitive to minor variations in its input). Generalization is useful because real-world data is noisy, distorted, and often incomplete. The ANNs ability to see through noise and distortion to the pattern that lies within is extremely important to pattern recognition in a real-world environment. Fourth, ANNs are also very good at the *abstraction* of essential characteristics from inputs containing irrelevant data. For example, a network can be trained on many distortions of the letter "A". After adequate training of the network, a perfectly formed letter will be produced. The end result is a network that has learned, through noisy and distorted examples, to produce something that it has not seen before. Fifth, they are *nonlinear*, that is, they can solve some complex problems more accurately than linear techniques do. Nonlinear behavior is quite common, but can be difficult to handle mathematically. Finally, ANNs are *highly parallel*. They contain many identical, independent operations that can be executed simultaneously, often making them faster than alternative methods. These abilities are a distinct asset in many applications because it does not require conventional programming, and while knowing something about the problem at hand can improve the neural network design (i.e. which inputs to include in the network training) the most important need is for data.

ANNs also have several drawbacks for some applications. Firstly, they may fail to produce a satisfactory solution, perhaps because there is no learnable function or because the data set is insufficient in size. Secondly, the optimum network geometry (i.e. the number of hidden layers and the number of hidden nodes per hidden layer) as well as the optimum internal network parameters (i.e. learning rate and momentum term) are problem dependent and generally have to be found using a trial-and-error-process. This can make the neural networks training process slow and expensive. Part of this cost comes from the need to collect, analyze, and manipulate training data while the other costs stem from the need to experiment with parameters to find good values. *Hammerstrom* [1993] believes that a greater understanding of training protocols and expert knowledge of the parameters which govern the process should help, especially in combination with parallel hardware that can run systematic experiments quickly. Another disadvantage of ANN models is, unlike ARMA-type models, a direct mathematical expression, relating the variables, is not obtained. Instead, the relationship between the inputs and outputs is contained in the connection weights, therefore making it hard to account for a neural network's results. Showing how the weights "cause" a result may be more complex than showing how a computer program works. ANN models are also problem specific. Physically based models, on the other hand, have had a wider range of applicability because they are based on fundamental physical relationships. Physically based models are also able to cope with changes in the system being modelled. ANNs cannot cope with major changes in the system because they are trained on a historical data set and it is assumed that the relationship learned will be applicable into the future. If there are any major changes in the system, the neural network would have to be quickly adjusted to the new process through model retraining, in which new data describing the process are added to the network's learning procedure.

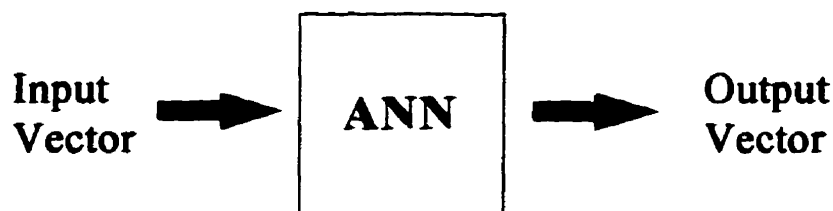
As previously mentioned, neural networks belong to a class of *data-driven* approaches, as opposed to traditional computer *model-driven* approaches. These models attempt to achieve good performance through dense interconnections of simple computational elements. In this respect, the architectures of ANNs are based on the present

understanding of the biological nervous systems. The highly sophisticated human brain, which contains more than 100 billion neurons and trillions of interconnections, is able to learn quickly from experience and is generally superior to any existing machine in tasks involving recognition, learning, and control [Hsu et al., 1995]. Although the present understanding of ANNs are based on that of biological neural networks, the structures of the most current ANNs are extremely simple and the capabilities are quite poor when compared to biological neural networks. Nonetheless, many ANN structures have been proposed and explored since the 1950s. The most widely researched and used structures are single and multilayer feed forward networks, Kohonen's self organizing feature maps, the Carpenter/Grossberg classifier, Hopfield networks, Hamming networks and Counterpropagation networks. Lippmann [1987] and Wasserman [1989] provide a brief summary and understanding of each of these network types. Of these, multilayer feed forward networks have been found to have the best performance in solving function approximation problems [Hsu et al, 1995]. Freeman and Skapura [1991] states that "*this model has a well-deserved reputation for being the neural network. It is a universal function approximator. We can, theoretically at least, teach anything learnable to this network. Thus, it is reasonable to consider this network for any problem*". Note that the network is said to be able to solve a function approximation problem if it is able to learn to approximate the function to arbitrary accuracy. Freeman and Skapura [1991] state that "*A multilayer feedforward network CAN learn your function. If you have problems, they are NOT due to the model itself. They are due to insufficient training, on insufficient number of hidden neurons, or an attempt to learn a supposed function that is not deterministic*". A short description about multilayer neural networks along with the training algorithms are provided next.

### 3.2 ANN ARCHITECTURE

Generally speaking, all ANNs are vector mappers. That is, they map one vector space to another. As shown in Figure 3.2, an *input vector* is applied to the network; in response, the network produces an *output vector*. Each vector consists of one or more *components*, each of which represents the value of some variable (e.g. precipitation, temperature,

streamflow, etc.). Mapping may be static, in which case a feed-forward neural network will suffice. Or, they may be dynamic, involving previous network states; in this case a network with feedback is required.



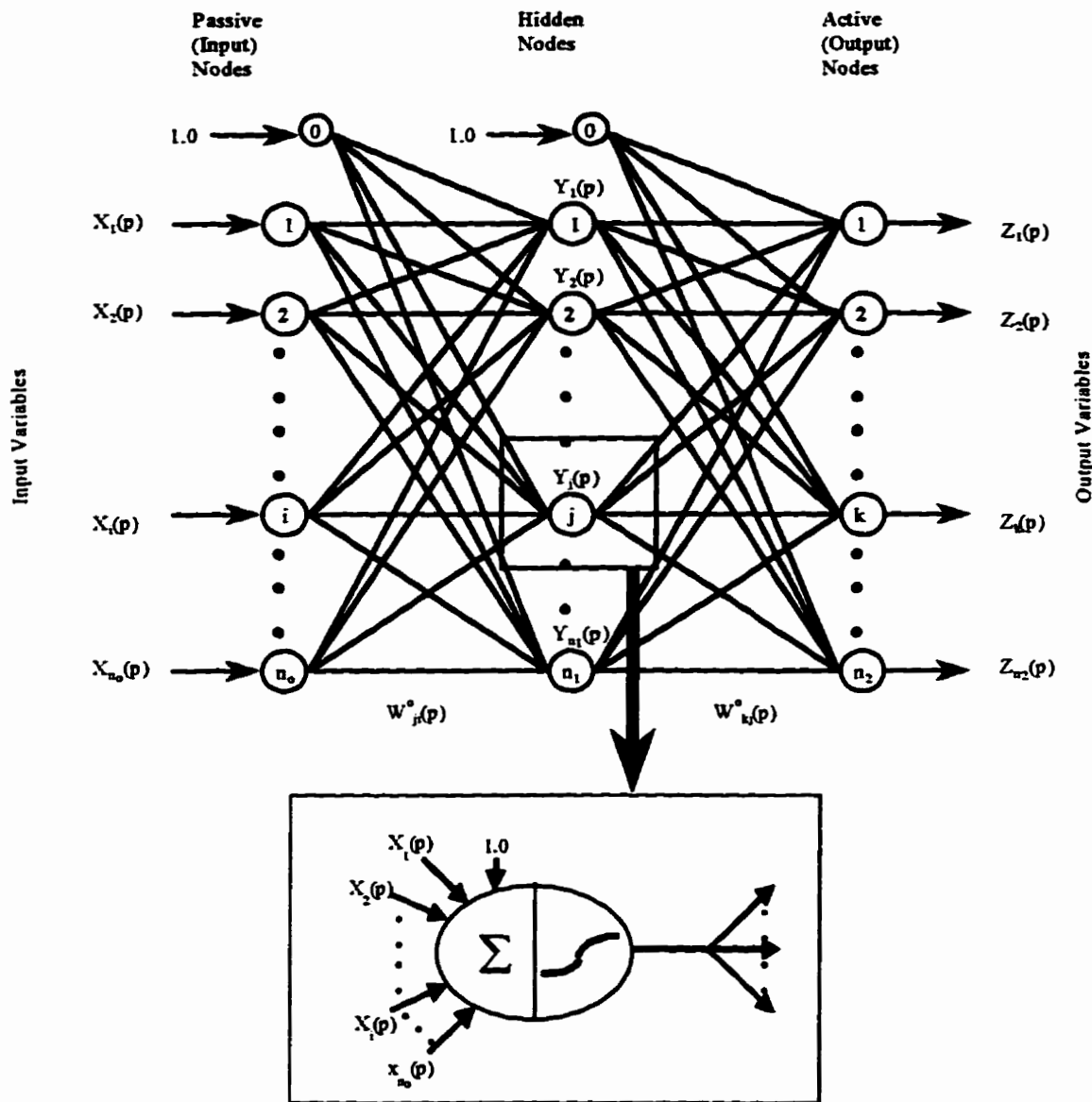
**Figure 3.2.** A Vector Mapper

### **3.2.1 Multilayer Feed Forward ANN Architecture**

The architecture of a feed forward ANN can have many layers. A typical three-layer feed forward ANN is shown in Figure 3.3. The first layer consists of a set of processing elements (PEs) and connects with the input variable(s). This is called the input layer. The last layer, consisting of PEs, connects to the output variable(s) and is called the output layer. The term “feedforward” means that information flows in one direction only. The inputs to neurons in each layer come exclusively from the outputs of neurons in previous layers, and outputs from these neurons pass exclusively to neurons in following layers. Layer(s) of PEs in-between the input and output layers are called hidden layers because they have no direct connection to the outside world, neither input nor output. There can be more than one hidden layer in an ANN. There are two conventions in use for counting the number of layers in the network; some authors count the input terminals as a layer, some do not. For purposes of this research, all layers are counted such that a network with one hidden layer is a three-layer network. Introducing this intermediate layer enhances the network’s ability to model complex functions. *Lippmann* [1987] suggests that no more than three layers (one hidden layer) are required in feed-forward networks because a three-layer network can generate arbitrarily complex decision regions. *Freeman and Skapura* [1991] believe that three layers (one hidden layer) are generally sufficient, however, sometimes a problem seems to be easier to solve with more than one hidden



layer. In this case, *easier* means that the network learns faster. *Valluru and Hayagriva [1993]* state that a feed-forward neural network with at least one nonlinear hidden layer of unlimited size, is a universal approximator, capable of approximating to an arbitrary degree of accuracy any continuous function on a compact set.



**Figure 3.3.** Multilayer Feed-Forward ANN and Processing Element Architectures

*Masters [1993]* has found that the number of hidden layers needed to solve most problems are as follows;

- 1) *"If the function consists of a finite collection of points, a three-layer network (one hidden layer) is capable of learning it"*.
- 2) *"If the function is continuous and defined on a compact domain, a three-layer network is capable of learning it. Roughly speaking, "compact domain" means that the inputs have definite bounds, rather than having no limits on what they can be"*.
- 3) *"Many functions that do not meet the above criteria can also be learned by a three-layer network. In particular, discontinuities can be theoretically tolerated under all conditions likely to be met in real life. Also, functions that do not have compact support, such as when the inputs are normally distributed random variables, can be learned by a three-layer network under some conditions"*.
- 4) *"Under very general conditions, all other functions that can be learned by a neural network can be learned by a four-layer (two hidden layer) network"*.

The first two cases above cover the majority of practical problems. Theoretically speaking, we are always reasonably safe using a single hidden layer. Furthermore, (at least theoretically) more than two hidden layers should never be needed. A network having two hidden layers is a universal approximator. The need for a second hidden layer can come about in essentially only one way. That is, when the network needs to learn a function that is mostly continuous, but has a few discontinuities. There are some functions defined on a compact domain that are generally continuous, but have one or more sudden jumps where the continuity is lost. These are called *piecewise-continuous functions* and, in general, cannot be easily learned by a network having only one hidden layer. Two hidden layers are usually required.

The PEs in each layer are called nodes or units. The number of nodes in the input and output layers are decided based on the dimension of input and output vectors presented to the network for training. The number of hidden layers and hidden layer nodes are decided upon by a trial-and-error procedure as there are no exact methods available in the literature. These nodes are connected to the nodes in the preceding layer for input and the following layer for output. There is no interconnection between nodes within the same layer or nodes in nonadjacent layers. Furthermore, each connection has an associated adjustable parameter called a weight or connection strength. An input connection may be excitatory or inhibitory. Excitatory connections have positive weights and inhibitory connections have negative weights. All weighted connections are “feed-forward”; that is, they allow information transfer only from an earlier layer to the next consecutive layer(s). Data enters the network through the nodes in the input layer. These nodes are passive, not computational, and simply serve as distribution points. They perform no input summation, but broadcast a single data value over weighted connections to the hidden nodes in the following layer. All of the hidden nodes receive all input data, but because each has a different set of weights, the sets of values differ. The number of hidden nodes must be large enough to form a decision region that is as complex as is required by a given problem. It must not, however, be so large that the many weights required cannot be reliably estimated from the available training data [Lippmann, 1987]. A network with too many weights may fit the training set well, but may produce large errors at intermediate points. In other words, the network will memorize the training set rather than generalizing around it. Conversely, networks with too few weights will fit both data points and the underlying function poorly. The following paragraphs describe the processing element (PE) architecture and transfer function.

### **3.2.2 The General Processing Element**

The architecture of a single PE or node is shown back in Figure 3.3. It has  $n$  inputs, labeled from 1 through  $n$ . In addition, it also has one assumed input, called its *bias*, who's input is always equal to 1.0. Each node  $j$  receives incoming signals from every node  $i$  in

the previous layer. A weight ( $w_{ji}$ ) is associated with each incoming signal ( $x_i$ ). The effective incoming signal ( $NET_j$ ) to node  $j$  is the weighted sum of all incoming signals, otherwise known as the net input:

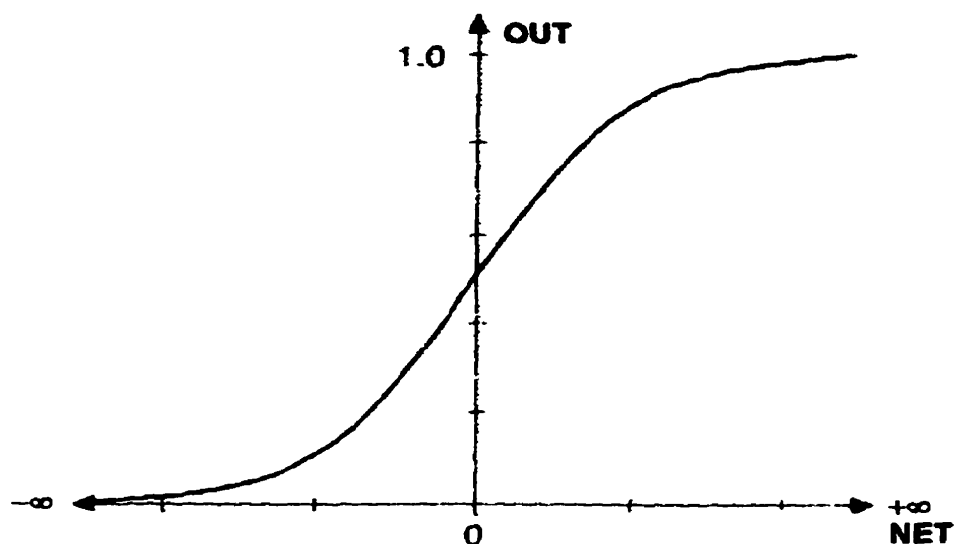
$$NET_j = \sum_{i=0}^{n_j} w_{ji} x_i \quad \dots 3.1$$

where  $x_0$  and  $w_{j0}$  are called the bias term ( $x_0 = 1.0$ ) and the bias weights respectively. This term is a weight on a connection that has its input value always equal to one. Including the bias term sometimes helps convergence of the weights to an acceptable solution. It is perhaps best thought of as an *extra degree of freedom*, and its use is largely a matter of experimentation with the specific application [Freeman & Skapura, 1991]. Note that excitation and inhibition are accounted for automatically by the sign of the weights. This weighted sum is passed through a transfer function and produces the neuron's output which serves as input to the next layer of nodes.

### 3.2.3 Transfer Functions

The transfer function, activation function, gain function, or squashing function (as they are often referred to) introduces a nonlinearity that, when applied to the net input of a neuron, determines the output of that neuron. This nonlinearity further enhances the network's ability to model complex functions. Nearly always, the same transfer function is used for all neurons. Wassermann [1990] states that the nonlinear activation functions are vital to the expansion of the network's capability beyond that of the single-layer network. The effective incoming signal,  $NET_j$ , is passed through a nonlinear activation function or transfer function to produce the outgoing signal/activation value ( $OUT_j$ ) of the node; see Lippmann [1987] for a list of three different activation functions. The outgoing signal of a node ( $OUT_j$ ), corresponds to the firing frequency of a biological neuron. The connection weights ( $w_{ji}$ ) and ( $w_{kj}$ ), between two nodes, corresponds to the strength of the synaptic connection between neurons in a biological network. Hsu et al. [1995] states that the most commonly used activation function is the steadily increasing S-shaped curve called a sigmoidal function. The shape of this curve can be seen below in Figure 3.4. This

function acts as a squashing function and compresses the range of  $NET$  so that  $OUT$  lies between 0 and 1. The desirable characteristics of the sigmoid function are that it is bounded above and below (the attenuation at the upper and lower limbs of the “S” constrains the raw sums smoothly within the fixed limits of 0 and 1), it is monotonically increasing, and it is continuous and differentiable everywhere.



**Figure 3.4.** Sigmoidal Logistic Function (Freeman & Skapura, 1991)

The sigmoidal function most often used for ANNs is the logistic function (*Hsu et al.* 1995), expressed mathematically as shown in equation (3.2). It can be seen that the activation value of any node is an explicit function of the net input value,  $NET_j$ , of that node;

$$OUT_j = f(NET_j) = 1 / \{1 + \exp(-NET_j)\} \quad \dots 3.2$$

in which  $NET_j$  can vary between  $\pm \infty$ , but  $OUT_j$  is bounded between 0 and 1. The activation function defines a nonlinear gain for the artificial neuron. The gain describes how the output leaving a node is changed based on the weighted sum entering the node. It is calculated by finding the ratio of the change in  $OUT_j$  to a small change in  $NET_j$  as shown in equation (3.3).

$$\frac{\partial OUT}{\partial NET} = OUT(1 - OUT) \dots 3.3$$

Therefore, the gain is the slope of the sigmoid curve at a specific excitation level. As can be seen in Figure 3.3, the gain varies from a low value at large negative excitation, to a high value at zero excitation, and it returns back to a low gain when the excitation of the neuron becomes very large and positive. Another commonly used activation function is the hyperbolic tangent, expressed mathematically in equation 3.4;

$$OUT_j = \tanh(NET_j) = \frac{e^{NET_j} - e^{-NET_j}}{e^{NET_j} + e^{-NET_j}} \dots 3.4$$

This function is similar in shape to that of the logistic function. The hyperbolic tangent function is also “S” shaped, but is symmetrical about the origin, resulting in  $OUT_j$  having a value of zero when  $NET_j$  is zero. A characteristic of the hyperbolic tangent function that is different from that of the logistic function is that the value for  $OUT_j$  is bipolar (i.e. the value of  $OUT_j$  can range between  $\pm 1$ ). This characteristic can be beneficial in certain networks whose outputs have negative values. It must be kept in mind that, regardless of what type of sigmoidal function is being used, the sigmoid functions never reach their theoretical minimum or maximum. For example, neurons that use the logistic function should be considered fully activated at around 0.9, and turned off at about 0.1 or so. It is certainly reasonable to use the extremes of 0.0 and 1.0 as inputs to a network, but it is futile to attempt to train a network to achieve extreme values as its outputs.

### 3.3 NETWORK CONFIGURATION

After choosing the ANN structure, the next step is configuring it to agree with the data. This is done by setting the number of input and output nodes to agree with the dimension of input-output vectors in the data set as well as deciding on the number of hidden layers and the number of hidden nodes per layer. The latter is an iterative process and will be

discussed further in the following sections. Since the number of inputs used in the network will dictate how many nodes are required in the input layer, it is important to decide which inputs are best in describing the underlying process. This is where familiarity with the application is invaluable. The raw data can also be inspected by means of statistical techniques, an example being calculating the strength of the correlation between an input and an output. This will suggest whether to include or exclude a variable in the input. Similarly, a strong correlation between two inputs might suggest that only one is needed. Data analysis helps screen out the potential input variables so that only the most telling ones are used to build the training patterns [Hammerstrom, 1993]. Preprocessing or transforming the data often makes it easier for the network to learn. Preprocessing may involve calculating sums, differences, differentials, inverses, powers, roots, logarithms, averages, and moving averages. One important reason for taking care in selecting both the type and number of input/output variables is to keep the network small, so that less time and data are needed to train it.

### 3.4 NETWORK TRAINING

The main objective of training a network is to produce the desired set of outputs when a set of inputs is fed to the ANN. This step is often called the calibration stage in conventional model building. Training a network is a process during which an ANN passes through a training set (input-output data pairs) repeatedly, changing the values of its weights (according to a predetermined procedure or algorithm) to improve its performance. It is important that the training set provide a full and accurate representation of the problem domain: otherwise the network will not meet expectations. A critical goal during training is to find a network that is large enough to *learn* the application but small enough to *generalize* well [Hammerstrom, 1993]. The number of nodes in a network has a large impact on the generalization ability of the network. Networks with too many hidden nodes tend to memorize the training data while those with too few cannot learn the underlying process in the problem. A network with many nodes will also have many connecting weights. Excessive weights can be a drawback, since the ANN can use them to memorize the training data. A smaller network has fewer

weights, forcing it to learn the underlying function and permitting it to generalize beyond the training data, but too small a network cannot learn the problem at all. Therefore, choosing the number of hidden layers and the number of hidden nodes per layer is an iterative process. Each pass through the training data is called an *epoch*, and the ANN learns through the overall change in weights accumulating over many epochs. During training, the network weights gradually converge to values such that each input vector produces output values that are as close as possible to the desired output vector. There are many algorithms available to train neural networks, but unfortunately no-hard-and-fast rules for matching training algorithms to applications exist. *Valluru and Hayagriva* [1993] estimate that over 80% of all neural network projects in development use the backpropagation (BP) training algorithm. This is possibly due to the relative ease of use of the algorithm. BP algorithms have been found to do particularly well at function approximation and time-series tasks as well as representing complex, nonlinear relationships in the form of compact, efficient networks [*Hammerstrom* 1993]. Before the training process is begun, all of the connection weights must be initialized to small random numbers (the size of the initial random weights is very important). This is to ensure that the network is not saturated by large values of the weights. If the initial weights are too large the sigmoids will saturate from the beginning, and the system will become stuck in a local minimum near the starting point. *Hertz et al.* [1991] suggests that a sensible strategy is to choose the random weights so that the magnitude of the typical net input,  $NET_j$ , to unit  $j$ , is less than, but not too much less than, unity. Typical initial weight randomization is between  $\pm 1$ .

The key to the BP training algorithm is its ability to change the values of the connection weights in response to the error between the network output and the target output. It accomplishes this through the use of the Generalized Delta Rule, which is the learning algorithm of the network.



### 3.4.1 Backpropagation Training Algorithm

The BP algorithm gives a prescription for changing the weights,  $w_{ji}$ , in any feed-forward network to learn a training vector of input-output pairs. It is a supervised learning method in which an output error signal is fed back through the network, altering connection weights so as to minimize that error. During the BP training, each input vector is applied to the first layer nodes in the input layer. This stimulus is propagated through each hidden layer and a result is generated at each output node. In forecasting applications, the training data usually consists of input stimuli as well as the desired responses (output) to these stimuli. During supervised training, the output predicted by the network,  $Z_k(p)$ , is compared with the actual (desired, historical) output,  $d_k(t)$ , and the mean squared error (MSE) between the two is calculated. The error signal, at time  $t$ , for each unit in the output layer, is given below by the error measure or cost function  $E(t)$ ;

$$E(t) = \frac{1}{2} \sum_k (Z_k(p) - d_k(t))^2 \quad \dots 3.5$$

The aim of training is to find a set of connection weights that will minimize the error function.

### 3.4.2 Generalized Delta Rule

The Generalized Delta Rule (GDR), used as the learning algorithm for the BP network, performs gradient descent down an error surface. Once the error signal of each node is known, the weights can be adjusted and updated in the direction that will minimize this error. This weight update is done with the GDR, shown explicitly in equation 3.6, where  $\alpha$ , is called the momentum term.

$$w_{ji}(t+1) = w_{ji}(t) + \Delta w_{ji}(t+1) + \alpha \Delta w_{ji}(t) \quad \dots 3.6$$

There are several methods for finding the weight increment,  $\Delta w_{ji}$ , of which the gradient decent method is the most common [Maier and Dandy, 1996]. The gradient descent

method is a learning process that changes an ANNs weights to follow the steepest path toward the point of minimum error.

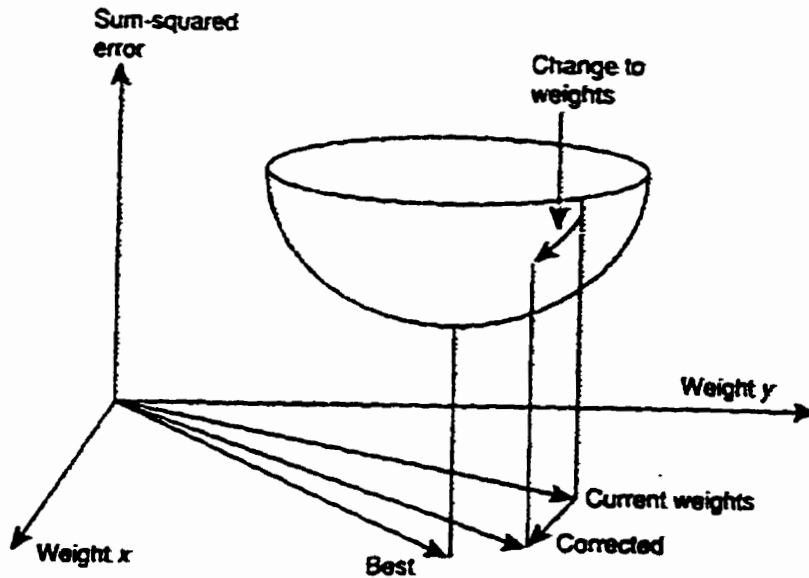
### 3.4.3 Gradient Descent Method

The change in the output errors ( $E$ ), with respect to the current weights ( $w$ ), are calculated (equation 3.3) and passed backwards (backpropagated) from the output layer to each node in the intermediate layer that contributes directly to the output. This is accomplished using the gradient descent method;

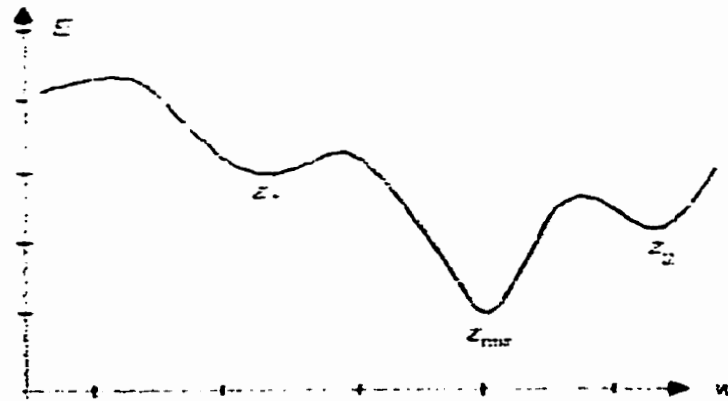
$$\Delta w_{ji}(t) = \eta \frac{\partial E}{\partial w} + \alpha \Delta w_{ji}(t-1) \quad \dots 3.7$$

As most combinations of weights produce a different error, an error surface exists as a function of the connection weights. To see this error surface, all possible sets of weights are plotted against the corresponding sum-of-squared errors. An error surface with two-dimensional weights ( $x$  &  $y$ ) is shown in Figure 3.5. This error surface is bowl-shaped whose bottom marks the set of weights with the smallest sum-of-squared errors. The gradient descent method results in weights being incrementally changed in the direction of the locally steepest path towards the bottom of the bowl.

Real error surfaces generally have a large number of local minima resulting in complex ravine-like features and many have dent-like local minima [Hammerstrom, 1993]. This local minimum is a point of regionally low error during gradient descent. It is a metaphorical dent in the error surface. Since the gradient descent method follows the steepest path, the network could train itself into a local minimum, ( $Z_1$  or  $Z_2$ ), that it can not escape from as in Figure 3.6.



**Figure 3.5.** Bowl-Shaped Error Surface (Hammerstrom, 1993)



**Figure 3.6.** A cross-section of a hypothetical error surface in weight space (Freeman & Skapura, 1991)

The global minimum is a unique point of least error during gradient descent. It is metaphorically the true “bottom” of the error surface. In practice, finding the global minimum ( $Z_{\min}$ ) is not always necessary. What is important is to find a set of weights that locate a local minimum that satisfy the accuracy requirements for the application at hand.

The size of the step taken down the error surface, often called the learning rate,  $\eta$ , can be critical. The learning rate governs the rate at which the weights are allowed to change at any given presentation. Higher learning rates speed the convergence process, but may cause the weights to oscillate around the bottom of the bowl-shaped weight surface, giving an overall error that is unacceptable. This can result in non-convergence of the problem. Lower learning rates slow the learning process resulting in more reliable results, but can increase the training time. In this case, convergence will be excessively slow. There are a number of ways of dealing with this problem, including the replacement of the gradient descent by more sophisticated minimization algorithms, as discussed by *Hertz et al.* [1991]. However, a much simpler approach, that is commonly used and often effective, is the addition of a *momentum term*,  $\alpha$ , expressed mathematically in equation 3.7

The momentum term must be between 0 and 1; a value of 0.9 is often chosen [*Hertz et al.*, 1991]. The momentum term is applied to prevent the search direction from wildly thrashing about the error surface, as the gradient changes. The idea of the momentum term is to give each weighted connection,  $w_{ji}$ , some momentum, so that it tends to change in the direction of the average downhill “force” that it feels, instead of oscillating widely. Each new search direction is computed as a weighted sum of the current gradient and the previous search direction. This momentum is derived from the effect of previous weight changes on the present change in weight space. This term is optional, but if used, can improve the convergence on a local/global minima. *Masters* [1993] states that the momentum term is essentially a low-pass filter applied to the search direction to dampen the side-to-side oscillations. The idea being that if rapid local fluctuations are filtered out, the remaining trend will be towards a more global minimum. The effective learning rate,  $\eta$ , can then be made larger without divergent oscillations occurring.

Each unit in the intermediate layer receives only a portion of the total error signal, based roughly on the relative contribution the unit made to the original output. This process repeats itself, layer by layer, until each node in the network has received an error signal that describes its relative contribution to the total error. Each hidden unit then calculates

the weighted sum of the backpropagated errors to find its indirect contribution to the known output errors. After the error value is calculated for each output node and hidden node, the network adjusts its weights to reduce its contribution to the total output error. For a more complete treatment of backpropagation see *Wasserman* [1990].

### 3.5 NETWORK TESTING

Results that are measured with the training data say very little about the ANN's reliability in the application. The only way to determine if an ANN is useful is if it produces appropriate results with data that was not used to train it. Measuring the generalization ability of the network requires testing the network with an independent set of data (i.e. data that was not used in training). The main objective of testing is to determine how the network performs on data it has not seen before. This step is often called the verification stage in conventional model building. Testing is a process for measuring an ANN's performance during which the network passes through an independent data set to calculate a performance index, such as the mean squared error (MSE), without changing the weights. The real objective during training is the test-set, not training-set, accuracy. Testing alternated with training is a good way to monitor the network performance and it shows when to end training to prevent memorization of the data, also called overtraining. A technique used to prevent overtraining is to stop when the mean-squared-error yielded by the testing set stops improving.

Test-set and training-set mean squared error both typically fall rapidly at the start of the training phase as the network moves its weights away from their random initial positions. As the training progresses, both the training-set and test-set error curves become flatter. Typically, training-set error continues to decline, but test-set error eventually begins to increase as shown in Figure 3.7. This type of behavior shows that the network has stopped learning what the training patterns have in common with the test patterns and has started to memorize meaningless differences. This overfitting of the training data, sometimes called overtraining, is caused by the network memorizing the noise in the training data. When a network is overtrained, it performs very well on the training data

but loses its ability to generalize on the test data. Therefore, for best generalization, training should stop when test-set mean-squared error reaches its lowest point. Training is therefore an iterative process: the trainer tries a configuration (number of hidden layers and nodes), evaluates the result (based on MSE), makes a change in the node configuration and parameters, and tries it again until satisfied with the result. There is an optimum number of hidden nodes producing the smallest error, with larger and smaller numbers of hidden nodes both producing larger errors. It is the experience of *Masters* [1993] that surprisingly few hidden nodes are usually needed. Monitoring error measured with a separate test set while varying the number of hidden nodes is a key strategy for optimizing architecture for performance with BP training.

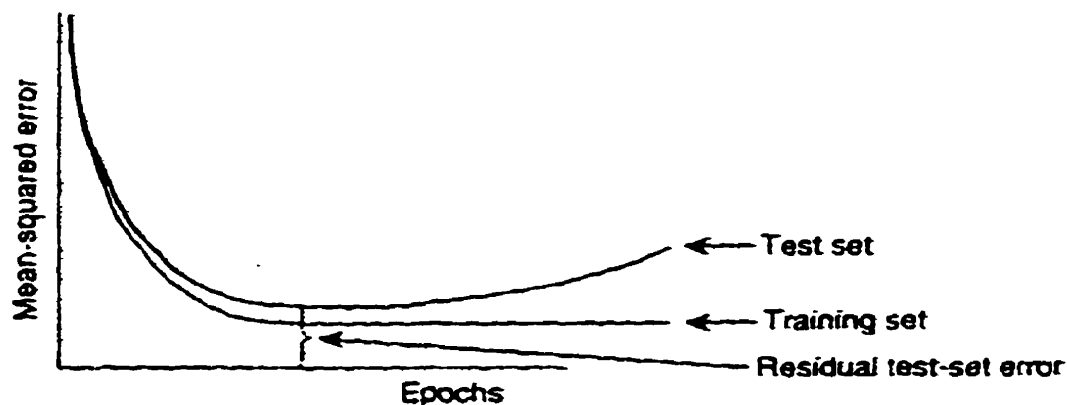


Figure 3.7. Training-set/Test-set error (Hammerstrom, 1993)

This trial-and-error process has hampered neural network development in the past, because it has often been impractical to try all possible node configurations and training parameters. There are many commercially available softwares, hardware, or combination thereof that use systematic methods to configure, train, and test ANNs to agree with the data. This makes the ANN technology more feasible for application to real-life problems. The network architecture (three fully connected layers) and the BP learning algorithm mentioned in the preceding sections provide a general idea about artificial neural networks and the most popular Backpropagation training algorithm. Not all of the details

mentioned above apply to all ANNs. Architectural elements such as layers, nodes, connections and weights are universal, but the arrangements between them vary. Some networks lack a hidden layer, and others have two or more. Some networks include nodes that feed back to earlier nodes (i.e. not feed-forward), connections that skip layers, and layers that are only partially connected. The learning algorithms controlling the updating of connection weights also vary for different ANN structures (i.e. the three-layer feed forward network described earlier might not use Backpropagation as its training algorithm and likewise, the Backpropagation algorithm might appear in network structures other than feed-forward networks as described in *Freeman and Skapura* [1991], *Hertz et al.* [1991], *Valluru and Halagriva* [1993] and *Wasserman* [1989]).

### **3.6 ANN MODEL DEVELOPMENT**

Designing an ANN can be as simple as selecting a commercially available software package and configuring it to agree with the data or as complex as coding a fully custom network from scratch - an approach beyond the scope of this research. The intended research concentrates on the application of ANNs for streamflow forecasting and not on the improvement or development of algorithms used in training ANNs. For this reason, conventional ANN model development (using Pascal, Fortran or C code) was not used as readily available commercial software packages, that have built-in training algorithms, can save time. The time saved on programming can be spent on preprocessing data and deciding which variables are driving the underlying process being modelled. Furthermore, this data analysis will help decide which variables to include as network inputs. Braincel™ [*Promised Land Technologies*, 1993] is the software that will be used to implement the ANNs in this research and will be discussed, in more detail in the case study chapter. Braincel™ is an Excel™ spreadsheet add-on program that is embedded in an Excel™ worksheet. Having the power and flexibility of the spreadsheet environment greatly enhances the power of neural network software. Braincel™ allows the user to fully access the formulas, functions, printing, and graphing facilities of the Excel™ spreadsheet environment. For these reasons, Braincel™ was the chosen software to be used in this

research. *Hammerstrom* [1993] gives a breakdown and comparison of ANN software packages currently on the market.

### **3.7 BRAINCEL (ANN tool)**

Braincel™ is one of the many readily available commercial ANN software packages on the market and as mentioned in chapter 3, is the software that was used to implement the ANNs in this research. Braincel™ is an Excel™ spreadsheet add-on program that is embedded in an Excel™ worksheet. The power of this ANN software is greatly enhanced by having the power and flexibility of the spreadsheet environment. Braincel™ allows the user to fully access the formulas, functions, printing, and graphing facilities of the Excel™ spreadsheet environment.

*“Braincel™ is great for forecasting and building expertise. In regards to forecasting, it is great at looking for relationships within data. Braincel™ should not be used when the formulas for a decision are already known and are fairly static. Instead, it should be used to solve problems for which it is difficult to formulate a procedural software solution or if the procedure is likely to change frequently. Braincel™ is an excellent tool for making discoveries in engineering and scientific disciplines”.*

*... .. Promised Land Technologies, [1993].*

Braincel™ uses the hyperbolic tangent as its transfer function and a variation of the backpropagation algorithm, called backpercolation (developed by Mark Jurik of JURIK RESEARCH AND CONSULTING), as its built-in training algorithm. This software allows the user to configure the number of nodes in both the input and output layers as well as configuring the number of hidden nodes in up to 7 hidden layers. The software automatically scales the input/output data sets as per the requirements of the transfer function used in the network. The only thing left to be automated by the user is how the network is to be trained and tested. The training and testing process can be automated by



Braincel™ in three ways: 1) Auto Expert Mode; 2) Automated Best Net Search; and 3) train on unseen data in Professional User Mode.

### **3.7.1 Auto Expert User Mode**

The Auto Expert Mode is the simplest of the three training modes available in Braincel™. In this mode, the user specifies the input/output structure, the hidden layer(s) structure and the data to be used for training and testing. The user then sets a timer to specify a time limit that the network is allowed to train for. In this mode, the user is not given much control over the parameters that affect training such as; randomization of initial connection weights (default value used in this training mode is  $\pm 1.5$ ) and learning rate ( $\eta$ ).

### **3.7.2 Automated Best Net Search**

In this mode, Braincel™ will automatically search out the best node configuration and weight randomization in a single hidden layer. Best Net will try out many initial randomizations of weights (between  $\pm 0.1$ ,  $\pm 0.5$  and  $\pm 1.5$ ) and hidden nodes. For example, if the user wanted the Best Net to check between 1 and 5 hidden nodes, the following steps would be taken:

- Step 1:** Check 1 hidden node with connection weights initialized between  $\pm 0.1$ . This sequence would be repeated 4 or 5 times before moving to the next step.
- Step 2:** Check 1 hidden node with connection weights initialized between  $\pm 0.5$ . This sequence would be repeated 4 or 5 times before moving to the next step.
- Step 3:** Check 1 hidden node with connection weights initialized between  $\pm 1.5$ . This sequence would be repeated 4 or 5 times before moving on to the next step, which would be to repeated steps 1 through 3 using 2 hidden nodes. These three steps would be repeated until all five hidden nodes were checked.

It is very easy for gradient algorithms to get stuck in local minima when learning feedforward network weights. *Chakraborty et al.* [1992] states that since backpropagation is a gradient-descent optimization technique, training a network may

result in its sinking into a local minimum that may be far removed from the global one. *Hertz et al.* [1991] suggest conducting several training runs for each network and states that...

*“we should always repeat the learning process from several different starting positions on the error surface. It is criminally reckless to use just one starting-weight configuration, assuming that the minimum to which it leads is the best that we can do”.*

This is exactly what the Best Net search is trying to accomplish by initializing weights between say  $\pm 0.1$  in Step 1, and then repeatedly setting the initial weights between  $\pm 0.1$  before moving on to Step 2.

The objective of the Best Net Search is to minimize the error on the data used for testing the network. It checks several, but not every, possible combination of hidden nodes and weights within the parameters set, before selecting another. Because this can take a lot of time, Braincel™ uses a smart algorithm to converge on likely combinations of hidden nodes and weights. Still, the larger the range of hidden nodes specified, the longer it's going to take. *Promised Land Technologies* [1993] suggest setting Best Net on one possible architecture at a time (i.e. select the same number as the minimum and maximum hidden nodes to be checked). By doing this, the Best Net will just experiment with initial weight randomization. When the Best Net search is finished, the optimum network is automatically saved and a message describing the optimum number of hidden nodes and initial weight randomization it has determined, is displayed.

One of the drawbacks to using the Best Net approach is that it only tests with one hidden layer. Some problems require two or more layers, but as was mentioned back in chapter three, these problems are quite rare. The Best Net also provides the user with a “*Leave Out Analysis*”. The Leave Out Analysis charts relative importance of inputs to the outputs and is used to; 1) give an indication of which variables are driving the underlying

function you are trying to approximate, and 2) which variables are of no importance and can be left out, for best training. *Promised Land Technologies* [1993] states that...

*“this ranking is not foolproof; it calculates which inputs are most sensitive in some parts of their range. Inputs that are steady throughout their range are undervalued with this method”.*

### **3.7.3 Professional User Mode**

Professional User mode is designed for experienced users of neural networks. In this mode, the user is able to gain manual access to many more parameters that make up the network such as the learning rate ( $\eta$ ) and the number of training cycles. In this mode, the user is responsible for a lot more of the parameters that affect how the network trains itself. Professional User Mode allows the user to manipulate several parameters of the neural network during training: displaying error on unseen data (test data) rather than on training data, counting training cycles rather than time (as in Auto Expert User Mode), monitoring and changing the learning rate, recording training history, and displaying an error chart while training progresses. To avoid overtraining or memorization of the data, the Professional User Mode has an Autostop feature that allows Braincel™ to stop training when the error on the unseen (test) data stops descending and begins to increase or reaches a plateau.

## **CHAPTER 4**

### **4 CASE STUDY APPLICATION**

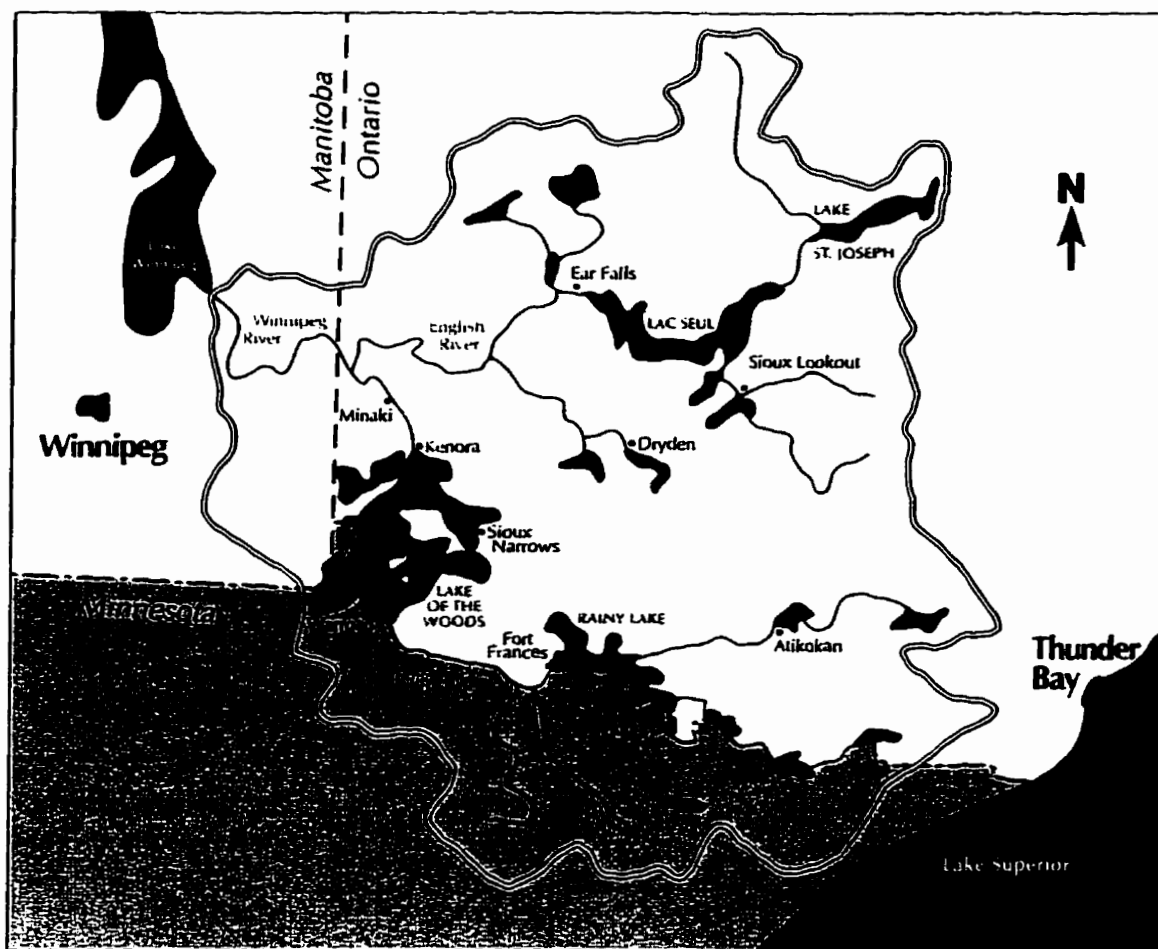
#### **4.1 GENERAL**

In this chapter, the use of ANNs for the real-time forecasting of streamflow in a portion of the Winnipeg River Basin located in Northwestern Ontario, Canada, will be described. This involves first separating the available data into a training set and a testing set. Through an iterative process of training and testing the ANNs, the best network configuration (number of input and output nodes, number of hidden layers and the number of nodes per hidden layer) is determined.

#### **4.2 DESCRIPTION OF TEST LOCATION**

The majority of the Winnipeg River Basin is located in the Precambrian Shield region of Northwestern Ontario with parts of the watershed entering Southeastern Manitoba. The Rainy River local drainage area, located in the Northeastern tip of Minnesota, is the only subwatershed, within the Winnipeg River Basin, that lies outside of the Precambrian Shield region. A map showing the location of the Winnipeg River Basin is shown in Figure 4.1. The Winnipeg River and its main tributary, the English River, constitute an enormous water resource, with a drainage area of approximately 150,000 square kilometers. There are more than 100 major lakes and rivers in the system, with a combined surface area of over 11 400 square kilometers [*Acres International Ltd.*, 1993]. For this reason, The Lake of the Woods Control Board (LWCB) was formed in 1919 and is one of several boards responsible for managing water in the Winnipeg River drainage basin. All runoff from the basin eventually makes its way down the Winnipeg River, spilling into Lake Winnipeg and then emptying into Hudson Bay. There are five regulated lakes in the Winnipeg River Basin (Lake St. Joseph, Lac Seul, Namakan Lake, Rainy Lake and Lake of the Woods) as shown in Figure 4.1. The basin consists of heavy woodlands, and countless lakes, rivers, swamps and waterfalls, thin soil and frequent rock outcrops

except for the portion of the basin that is in the state of Minnesota. This portion of the watershed has physiography and soil types that are very different from the rest of the basin.



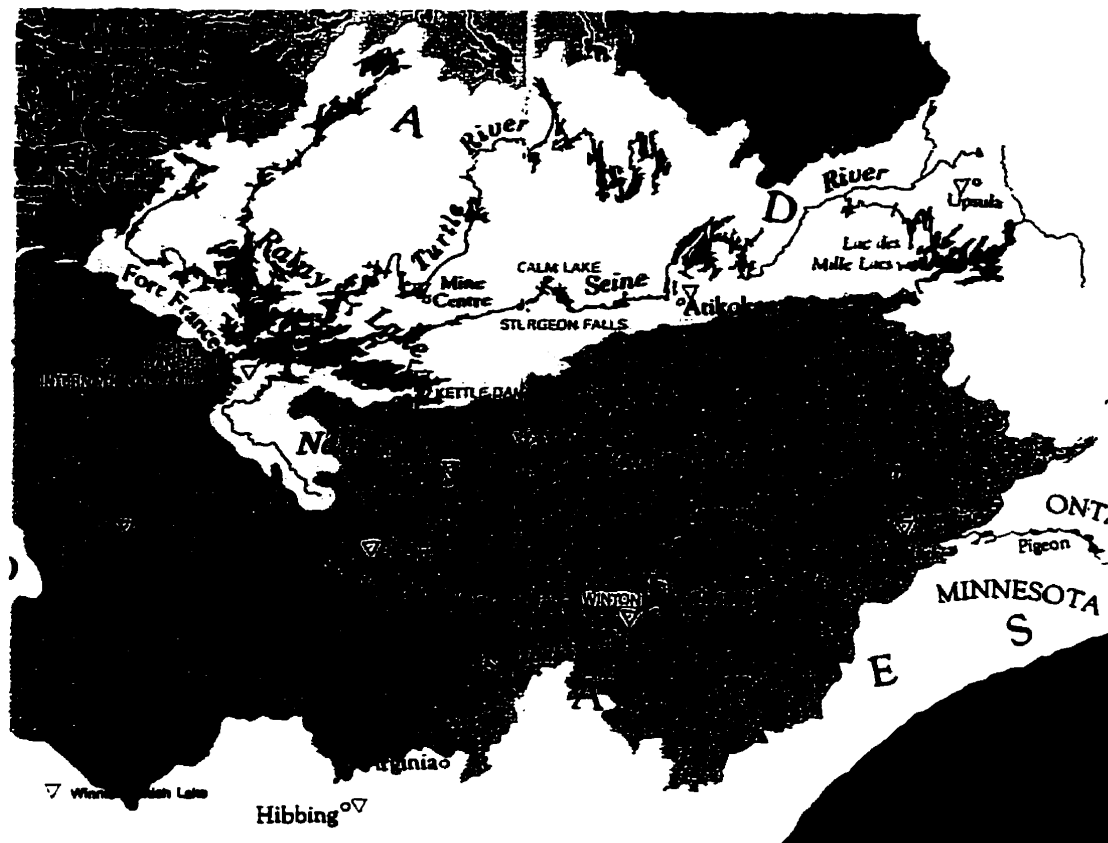
**Figure 4.1.** Winnipeg River Basin (LWCB brochure, 1994)

The basin consists of scattered settled areas that include many rural and recreation-oriented towns that are very active during the summer months. A significant portion of the economic base of the Winnipeg River Basin is closely tied to the water resources of the area. The area has a great diversity of water resource use in combining industrial, commercial, and recreational interests. Mineral and timber resources are still important elements of the local economy as evidenced by natural resource-based industries such as mining and pulp and paper. Tourism is a major industry in the region along with sport fishing and other outdoor recreational activities. The importance of water resources to the local economy has led to considerable concern regarding the proper management of the basin's water resources.

Specific to water resources concerns, serious flooding has been reported on the English River in the past with record-breaking water levels during 1992. In addition to flooding on the tributary rivers, two of the five major regulated lakes (Lac Seul and Lake of the Woods) have experienced problems during both high flow years, when flooding has resulted, and low flow years, when summer operating levels have not been reached on the lake. The problem of determining the optimal operating policy for the outlet controls in the Winnipeg River Basin has received much attention by the Lake of the Woods Control Board (LWCB) due to the economic importance of the five major lakes to the surrounding region, as discussed above.

The management of the Winnipeg River Basin system is particularly difficult due to interests in control of flooding, water based recreational activities, hydroelectric power generation, agriculture, and municipal and industrial water supply. Directing the operation of large lakes requires monitoring and forecasting basin conditions, planning regulation strategies, consulting with affected parties and providing public information. The unpredictability and variability of natural inflows, into the five major lakes, can make the LWCB's job extremely challenging. Since this research only focused on forecasting natural inflows, it was decided to model only the local inflow to one of these five lakes. The most upstream lake in the watershed, Namakan Lake, was chosen as the test location

in this case study. The Namakan Lake subwatershed has a drainage area of approximately 19 270 square kilometers. Figure 4.2 shows a schematic representation of the subwatershed and the locations of the control dams, water level, flow level and precipitation gauges. The information obtained from forecasted local inflows, into Namakan Lake, could be used to assist in modifications made to the normal operating procedure of the Namakan Lake subwatershed.



**LEGEND**

- Control dams -
- Water level and flow gauges •
- Precipitation gauges ▽

**Figure 4.2. Namakan Lake Subwatershed (LWCB brochure, 1994)**

### **4.3 WINNIPEG FLOW FORECASTING SYSTEM (WIFFS)**

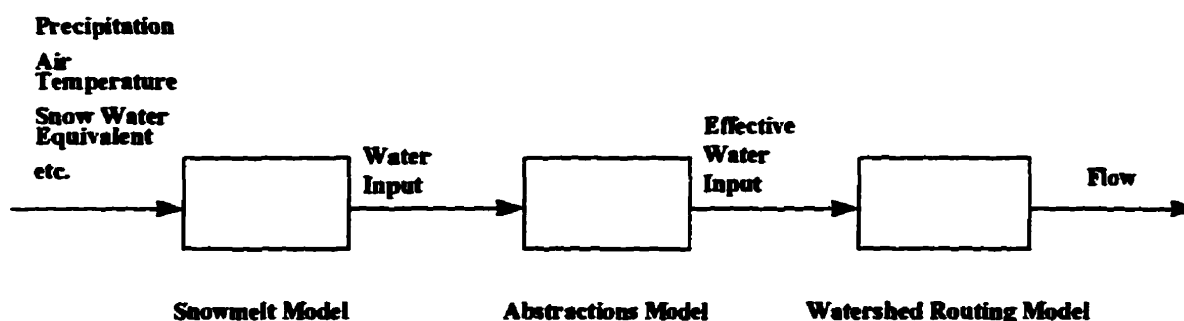
The LWCB is currently implementing a comprehensive water management model called the Acres Reservoir Simulation Program (ARSP), for reservoir operations in the basin. The robustness of this model is very important for proper management of the system. Forecasts of natural inflow for the next few quarter-monthly time intervals is of considerable value in using the ARSP model [*Acres International Ltd.*, 1993]. The definition of quarter-monthly time steps adopted by the Lake of the Woods Secretariat and used in the Winnipeg Flow Forecasting System (WIFFS) model includes four time steps each month. The first time step includes 8 days, the second and third 7 days, and the fourth has the remaining days in the month [*Lake of the Woods Control Board (LWCB)*, 1994]. This particular breakdown of periods translates into 48 periods per year. The WIFFS model was developed by ACRES International of Niagara Falls, Ontario, to estimate these quarter-monthly natural inflows into the major lakes in the system. The WIFFS model served as the comparison tool to the ANN method.

#### **4.3.1 Model Type**

There are a number of watershed models that have been developed and applied to operational forecasting. Most of these models (e.g. the Streamflow Synthesis and Reservoir Regulation (SSARR) model, developed by the US Corps of Engineers; and the Hydrologic Simulation Program - Fortran version (HPSF) developed by the Hydrocomp Incorporated) were initially developed for *simulation* purposes and are referred to herein as conventional watershed models. Other more recent watershed models, such as stochastic-deterministic models, were primarily developed for *forecasting* purposes. WIFFS is a model of the second type. This watershed model contains three basic components as shown in Figure 4.3. These include:

- (i) water input generation model;
- (ii) abstraction or loss model; and
- (iii) distribution or watershed model.





**Figure 4.3. Components of WIFFS Model**

The water input generation component includes a procedure to calculate rainfall and snowmelt water input to the watershed, based on the meteorological data. The abstractions model estimates the portion of the water input that does not infiltrate into the ground but rather becomes runoff. This runoff is commonly referred to as effective or direct water input. The watershed routing component of the model routes this effective water input through the watershed. In stochastic-deterministic watershed models (as in the WIFFS model) the persistence properties of streamflow and the covariance (correlation structure) between streamflow and water input are modelled using multivariate time-series models. In stochastic-deterministic models, the basic assumption according to *Acres International Ltd.* [1993], is that the forecasted flow can be considered to be the sum of a deterministic component and a stochastic component (noise model). The deterministic component of the model relates the current flow to past flows and current and past water inputs. The stochastic component (noise model) is included in recognition of the fact that forecast errors do occur and that these errors have some autocorrelation structure. For a more detailed description of each of the three components of the WIFFS model, the reader is referred to the report by *Acres International Ltd.* [1993] entitled "Winnipeg River Basin Flow Forecast Model Development".

### 4.3.2 Model Inputs

The WIFFS model employs a mixed time step, that is, precipitation and snowmelt water input is calculated using a daily time step and daily precipitation and temperature are required. Quarter-monthly effective water input is then calculated from the daily effective water input. The stochastic-deterministic watershed routing models use a quarter-monthly time step and quarter-monthly average local inflow is required.

The inputs to the WIFFS model are as follows:

- ***Period “now”,  $P(t)$*** ; this is the period before the first forecast period.
- ***Forecast lead-time***; is the number of periods from Period “now” for which forecasts are calculated.
- ***Historical Meteorological inputs***; complete daily precipitation and temperature data for Period “now” and the previous seven periods for Namakan Lake. The number of previous periods of meteorological data required may differ for each subwatershed.
- ***Forecasted Meteorological inputs***; an inference of future daily precipitation and temperature inputs. The WIFFS model includes two options for creating future meteorological inputs, i.e. probabilistic and manual entry of forecasts. Probabilistic creations of future meteorological inputs involve choosing a percentile of distribution to be used as the forecast and manual entry involves picking an actual value and entering it as the forecasted number. For purposes of this research (to make comparisons between modelling approaches easier) the probabilistic method, with the 50<sup>th</sup> percentile of distribution, was used in developing these future meteorological inputs. A more detailed discussion on this probabilistic method used to construct the future precipitation and temperature sequences can be found in *LWCB* [1994], and

- **Historical flows**; quarter monthly flow data for Period “now”,  $P(t)$ , and the previous seven periods for Namakan Lake. Again, the number of previous periods of flow data required may differ for each subwatershed.

The outputs from the WIFFS model are as follows:

- **Streamflow ( $t+1, 2, 3$  or  $4$ )**; One, two, three or four week(s)-in-advance forecast of local inflow. For the remainder of the thesis, each time period will be referred to as a “week”, even though each quarter-month period is not comprised of seven days.

#### 4.3.3 Available Data and Resources

The Water Survey of Canada (WSC) has operated hydrometric gauging stations for a number of sites in the Winnipeg River Basin. The locations of the six WSC gauges in the Namakan Lake subwatershed, for which there are flow records available, are shown in Figure 4.2. LWCB has, in recent years, developed an extensive network of data collection platforms (DCPs) making some of the meteorological data available in real time. The three stations that have historical meteorological data, in the Namakan Lake subwatershed, are shown in Figure 4.2 along with the two stations (Winton and Gunflint) that have the data available in real-time. Basin average precipitation was estimated for each subwatershed using 3 to 6 selected records from stations with historic and real-time data. Thiessen’s polygon method was used to determine the weighting coefficients for each record. The stations and coefficients used for each subwatershed are summarized in the report by *Acres International Ltd.* [1993]. There were twenty-nine years of available data from 1960-1988 for the Namakan Lake subwatershed. This data included average daily precipitation and temperature values as well as average weekly streamflow values. The historical data for Namakan Lake can be found in Appendix A.

The stochastic-deterministic watershed model (WIFFS) was calibrated by *Acres International Ltd.* [1993] for each subwatershed using daily precipitation and temperature data and quarter-monthly average flow data. Data from the period 1965 to 1985 were

used for model calibration, and the available data before 1965 and after 1985 were set aside for model verification purposes *LWCB* [1994]. It should be noted that the WIFFS model was not run as part of this research. Rather, the results (R.M.S.E) already compiled by *Acres International Ltd.* [1993] were used for comparison with the results from the ANN technology.

#### **4.4 TYPES OF MODEL COMPARISONS**

As mentioned earlier in chapter one, the ultimate evaluation of the ANN model came through a comparison of the network model with a traditional forecasting tool (WIFFS). The two modelling approaches will be compared not only in terms of the accuracy of the forecasting results but also in terms of the ease of use of the two approaches. The following two sets of experiments were performed in this study.

##### **4.4.1 Experiment #1**

In the first experiment, an ANN model was produced to give a fair comparison between the forecasting accuracy of the WIFFS model and that of the ANN technology. This was done by providing the ANN with the same inputs that were used in the development of the WIFFS model. The ANN model was then trained and tested on the same sets of data that were used to calibrate and verify the WIFFS model respectively. In this experiment, the ANN made use of all 29 years of data from 1960-1988, as did the WIFFS model.

##### **4.4.2 Experiment #2**

In the second experiment, an ANN that was not restricted to the input variables used in the WIFFS model, was built. Using too many inputs can excessively complicate the model, whereas using too few inputs can result in inadequate modelling. In the context of ANNs, *Chakraborty et al.* [1992] indicate that too many inputs would imply slower training and slower convergence and may in fact worsen the generalization capabilities (applicability to test cases) of the network. The purpose of the second experiment (called experiment #2.2) was to build an "optimum" ANN. This "optimum" network involved a smaller number of inputs (i.e. only inputs that were being used to describe the underlying

process being modelled), thus reducing training times, yet still remain as accurate if not more accurate than the ANN model built in experiment #1. This was done using some of the same inputs as in WIFFS and investigating the use of some new, additional inputs, that were not used in the WIFFS model. In this experiment, part of the data (from period 33 of 1983 to period 33 of 1984) had to be eliminated from the data set. The reason for doing this was that it appeared the actual measured data was missing. This missing data appeared to be artificially filled-in for purposes of keeping the time-series continuous. Therefore, to obtain a proper comparison between the accuracy of the ANN developed in experiment #1 and that developed in experiment #2, these data were eliminated and experiment #1 was redone and called experiment #2.1. The ANN model was then trained and tested on the same sets of data that remained to calibrate and verify the WIFFS model respectively.

## **4.5 TIME-SERIES FLOW FORECASTING USING ANNs**

### **4.5.1 Single Predictor**

Any neural network that is capable of accepting real-valued vectors as input and producing real-valued vectors as outputs may be used for time-series prediction. For now, we will assume the simplest possible structure (using lags of one particular variable as inputs to the network). The ANN is a black box having one or more inputs and *exactly* one output. We are attempting to predict one point ahead in a single time-series. Extensions to this simple structure will be discussed later. A typical mapping of a time-series to an ANN for prediction is outlined in Figure 4.4. This schematic demonstrates the use of six continuous points (of a single variable) in a time-series to predict the next point. The training series is produced by generating a large number of individual samples. Each sample consists of seven points: the current point (lag 0), five historical points (lags 1 through 5), and the following point (often called the target output), which is used to direct the training of the output neuron. It is possible to make use of more than one variable along with their lags as well. As can be seen, the training set can become very large. Figure 4.4 shows the prediction to be exactly one point into the future. Many times, as in

the case of this research, we need to predict further out than just one point. Multiple predictions from one network are possible, but are generally best avoided. *Hertz et al.* [1991] state that, it is best to limit a network to having to make only one lead prediction, whether that lead is one point into the future or one hundred points into the future. Although most networks can easily accommodate multiple outputs, *Hertz et al.* [1991] state it is better not to try to train multiple outputs to learn different lead predictions. Instead, train separate networks for each lead-time. In other words, a network should not be designed with two outputs, training one of the outputs to predict one point ahead, and the other output to predict one hundred points ahead. Instead, two separate networks can be used.

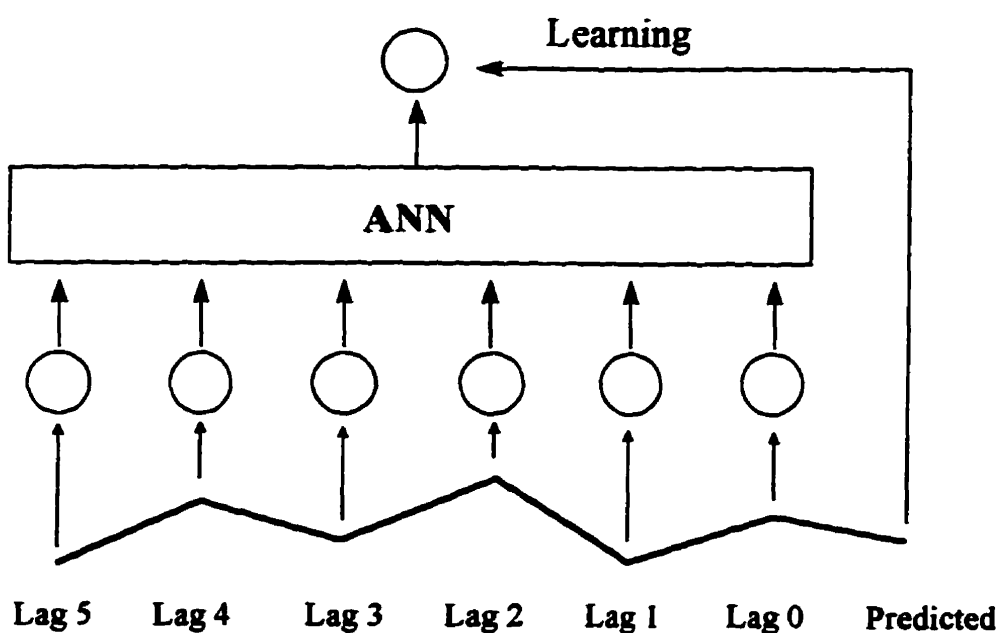
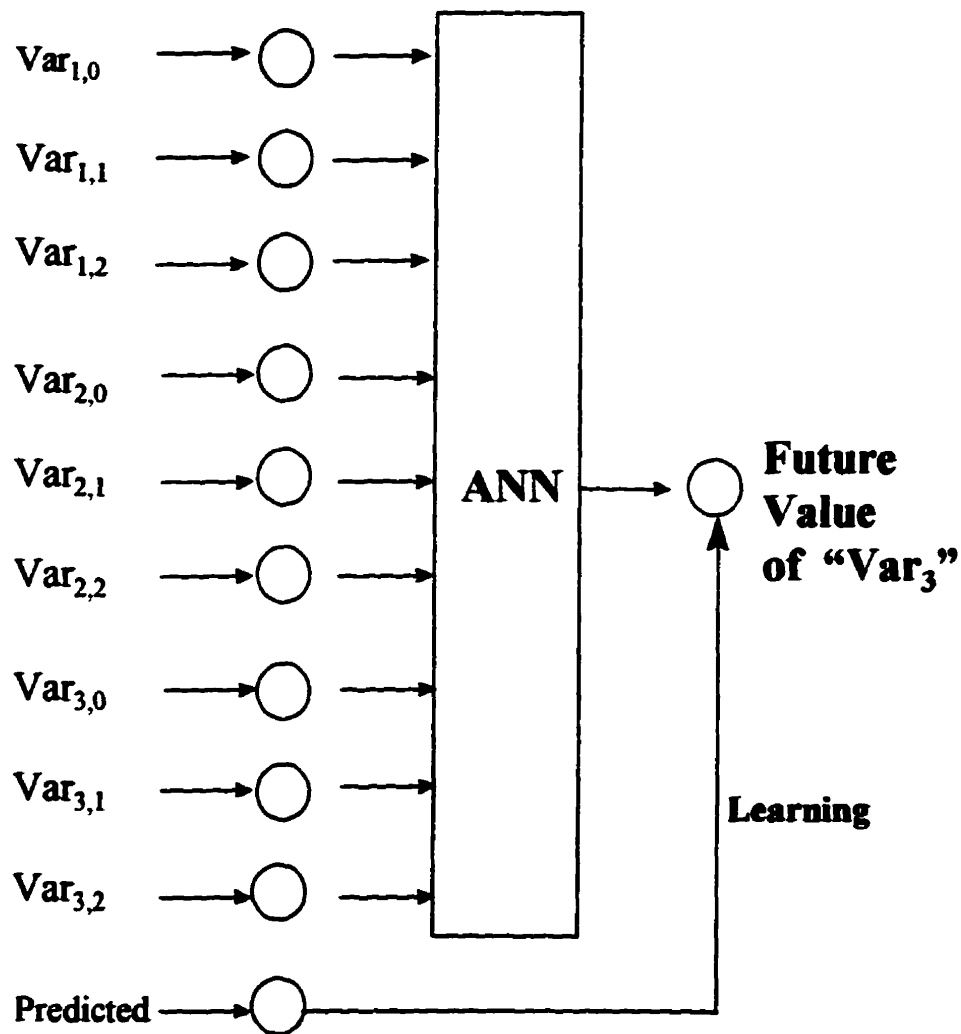


Figure 4.4. Univariate time-series prediction model.

#### 4.5.2 Multiple Predictors Series

The most basic time-series predictions use lags from a single-variable time-series as the predictors, as discussed in the previous section. If needed, this can be extended to multiple time-series of multiple variables as shown below in Figure 4.5. Figure 4.5 demonstrates the use of three lags of three different variables,  $Var_{i,j}$ , where  $i$  corresponds to the variable type and  $j$  corresponds to the lag of that variable type. The lags of these three variables create three separate time-series to predict the next point of  $Var_3$ . Again,

the training series is produced by generating a large number of individual samples. In this case, each sample consists of ten points (9 inputs and 1 output). The inputs are the current point (lag 0) of all three variable types and two historical points (lags 1 and 2) of all three variable types. The output is the following point (often called the target output) of the variables being predicted, which is used to direct the training of the output neuron.



**Figure 4.5.** Multivariate time-series prediction model.

The inputs to a time-series prediction model serve as information about the underlying relationship being modelled. This input information need not be limited to measured values. It is generally agreed [Hertz *et al.*, 1991] that, inputs that provide the network

with information about the underlying process being modelled, and are not restricted to time domains, are referred to as “identifying” information. An example of identifying information is in the recent work by *Ramesh and Mujumdar* [1996]. They used the “period of the year” as an input to their neural network for a 10 day ahead forecast of rainfall. The addition of this “identifying” information proved to be extremely useful in letting the neural network know what season of the year the current time step was in. This type of information is not easily incorporated into a single, traditional ARMA model.

Instead, separate ARMA models must be developed for each season of the year. It is important to remember that the guideline for neural network data is that if it is not clear whether a particular input variable will be of much use to the network, it should be included so that the network can decide what is important and what is not. It is important to note that the more inputs variables fed to the network, the more connection weights between neurons which result in a more complex error surface and ultimately longer training times.

## **4.6 ANN MODEL IDENTIFICATION**

In both experiments, a multivariate time-series model was derived for the output  $Flow_{t+i}$ ;  $i = 1, 2, 3,$  and  $4$ . The data used are total weekly precipitation, average weekly temperature and average weekly streamflow over the period from January 1, 1960 to December 31, 1988 obtained from *Acres International Ltd.* [1993]. In all cases, the networks were trained over a certain part of the data (1965-1985) and once training was complete, the networks were tested over the remaining data (1960-64 and 1986-88), i.e. to make the network predict the so-called “future values”.

### **4.6.1 Experiment #1**

The number of input and output variables used in the ANN model were fixed to match those used in the WIFFS model. These inputs included precipitation, temperature and local inflow of the preceding weeks as well as rainfall and temperature of the examined



week. For the one-week ahead forecast of local inflow,  $Flow(t+1)$ , the model consisted of the following 18 input variables in the time-series equation:

- past seven periods of weekly precipitation,  $P(t), P(t-1), \dots, P(t-6)$ .
- past seven periods of weekly temperature,  $T(t), T(t-1), \dots, T(t-6)$ .
- past two periods of local inflow,  $Flow(t)$  and  $Flow(t-1)$ .
- one week ahead inference of the average weekly precipitation,  $P(t+1)=50^{\text{th}}$  percentile, and
- one week ahead inference of the average weekly temperature,  $T(t+1)=50^{\text{th}}$  percentile.

The two week ahead forecast included an additional 3 inputs [ $P(t+2)=50^{\text{th}}$  percentile,  $T(t+2)=50^{\text{th}}$  percentile, and  $Flow(t+1)$  calculated in the one-week ahead forecast] for a total of 21 inputs and 1 output [ $Flow(t+2)$ ]. The number of input/output variables for the three and four week ahead forecasts followed the same sequence and included 24 inputs – 1 output and 27 inputs – 1 output, respectively.

All temperature values less than zero were set to zero before being submitted to the network for training. This eliminated any negative numbers in the data which made scaling of the data between 0 and 1 possible. The WIFFS model dealt with the temperature values in a similar way by setting a base temperature between  $-1.8^{\circ}\text{C}$  and  $0^{\circ}\text{C}$ .

Since the number of inputs (18) and outputs (1) for the one-week ahead forecast were fixed to match those of the WIFFS model, the input layer consisted of 18 input nodes with one node in the output layer. The only part of the network that remained to be configured was the hidden layer(s). *Cheng and Noguchi* [1996] state there is no well-defined algorithm for determining the optimal number of hidden layers and hidden nodes. Initial forecasting results indicated that the problem of forecasting streamflow in this River Basin could be accomplished with only one hidden layer. Networks were initially configured with two hidden layers but improvement in forecasting results were only marginal if at all,

with training times increasing from approximately ten minutes to approximately two hours. Therefore, the ANNs created for modelling the rainfall-runoff transformation in this subwatershed have been built considering a one-hidden layer ANN for each simulation trial. The problem then became that of how many nodes would make up this single hidden layer. *Cheng and Noguchi* [1996] indicate that selecting too many hidden neurons will increase the training time but without significant improvement on training results. *Ranjithan et al.* [1993] state that too many hidden neurons will encourage each hidden neuron to memorize one of the input patterns, and thereby diminish the interpolation (generalization) capabilities. Taking too few hidden neurons will decrease the training times but will restrict the network from generalizing on new data (test data) and therefore reduce the accuracy of prediction. Guidance as to how many hidden-layer nodes should be used is given by several authors. A number of these guidelines [*Baum and Hausler*, 1989; *Weigend et al.*, 1990] relate network size to the number of training samples available. However, as pointed out by *Chakraborty et al.*, [1992] such methods are too restrictive when a small number of training samples is available. *Hecht-Nielsen* [1987] suggests an upper limit of  $2I+1$  hidden layer nodes, where  $I$  is the number of inputs. However, in many practical applications, the number of hidden layer nodes was found to be considerably less [e.g. *DeSiltes et al.*, 1992]. *Maren et al.* [1990] indicate that for many applications, the optimum number of hidden layer nodes has been found to be less than the number of inputs. *Ranjithan et al.* [1993] state that the general practice is to determine the number of intermediate units by trial-and-error based on a total error criterion. However, several adaptive approaches have been recently reported by *Hirose et al.* [1991] and *Karnin* [1990]. These methods automatically add or remove intermediate nodes, driven by a total error criterion, during the training process. These approaches have been shown to be more efficient than the trial-and-error procedures but require these algorithms to be manually coded into a program. This process is beyond the scope of this research and the trial-and-error process was used. Therefore, training was done on ANNs with hidden layer nodes varying from 1 to 30.

Initial experimentation with the Braincel™ software indicated that the *Automated Best Net Search* would be the easiest to implement and would also produce more accurate results than those produced by the *Professional User Mode* and the *Auto Expert User Mode*. After reviewing the current literature, it became obvious that the most common mathematical means of measuring the effectiveness of each network structure was the Mean Squared Error (MSE). Therefore, the MSE was calculated and monitored for both the training and test sets of each network configuration. The systematic approach used in experiment #1 to identify the optimum hidden layer structure was as follows;

**Step 1:** Set the minimum and maximum number of hidden nodes to be checked by the Automated Best Net Search to 1 and 5 respectively.

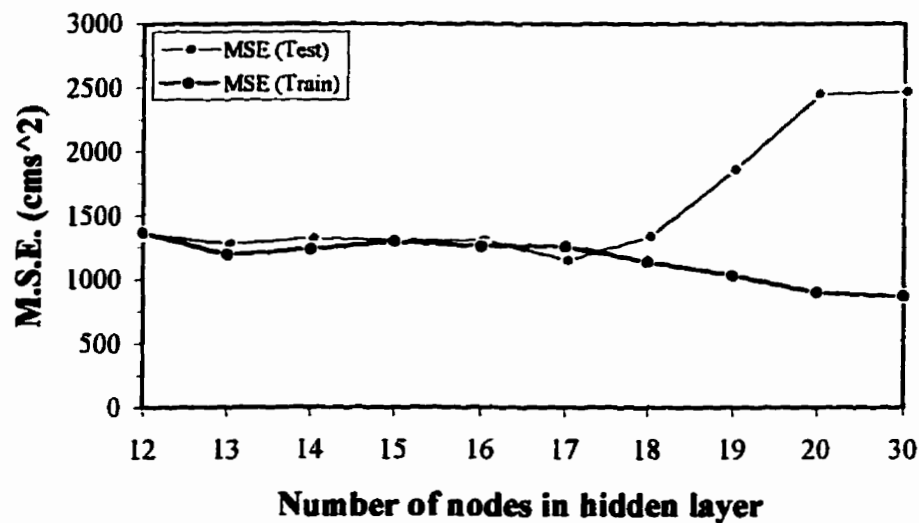
**Step 2:** Let the network self-train to completion and record the MSE for both the training set and testing set data.

**Step 3:** Repeat steps 1 and 2 by increasing the minimum and maximum number of hidden nodes to be checked, by 5 nodes respectively. Repeat steps 1 and 2 until the network has checked a hidden layer structure consisting of up to 30 hidden nodes.

**Step 4:** Plot the number of hidden nodes versus both the MSE(test) and MSE(train) as shown below in Figure 4.6.

The network with the lowest MSE (test) is termed the best network because it is the one that performs the best at generalizing on new data. Figure 4.6 indicates that the error of the training set decreases gradually with an increasing number of intermediate units. Further, the error predictions for the test set decreases initially but tends to increase after reaching a minimum. These observations indicate that although the training process becomes progressively easier, the generalization capabilities of the network reaches an optimum and does not improve indefinitely with an increasing number of intermediate nodes. This suggests that, for the one-week ahead forecast, networks with greater than 17 hidden nodes are oversized for the given multivariate data and a network with a single hidden layer consisting of 17 hidden nodes is suitable for prediction. This model structure

is represented by the notation  $ANN_{L_{\text{out}}}(n_i, n_h, n_o)$ , where  $n_i$  is the number of nodes in the input layer,  $n_h$  is the number of nodes in the hidden layer, and  $n_o$  is the number of nodes in the output layer ( $n_o = 1$  in our case). Therefore, the optimum ANN structure for the one-week ahead forecast is presented as  $ANN_1(18, 17, 1)$ . The optimum ANN structure for the two, three, and four-week ahead forecasts were as follows;  $ANN_2(21, 9, 1)$ ,  $ANN_3(24, 24, 1)$ , and  $ANN_4(27, 18, 1)$ .



**Figure 4.6.** Experiment #1, Variation of MSE for (1) training set and (2) testing set, with the number of hidden nodes for the one-week ahead forecast.

#### 4.6.2 Experiment #2

After training the initial model in experiment #1 with the same inputs used in the WIFFS model, two new inputs were added to the data set. These additional inputs, referred to as “identifying information”, proved to be helpful in training the network. The first of these two inputs was the “period of the year”,  $Period(t+1)$ , of the one-week ahead forecast. This input provided the network with information on which season the forecast was being made in. The second of the two inputs was the cumulative precipitation from November 1<sup>st</sup> to the time of the current period of the year,  $Cprecip(t)$ , up to April 1<sup>st</sup>. This input represented a measure of the amount of snowpack that accumulates over the winter and

adds to the spring runoff. This input was particularly helpful in accurately forecasting the rising limb of the hydrograph during the spring runoff periods. As in experiment #1, only one hidden layer was used in the network with the problem again of how many nodes to use in the hidden layer. In addition to determining the number of nodes in the hidden layer was the problem of how many nodes should be used in the input layer. Inputs were not being restricted to only those used in the WIFFS model and inputs that were not being utilized by the network to learn the underlying function being modelled were eliminated.

With the additional two inputs added to the data set, a sensitivity analysis of all of the input variables was carried out to determine the relative significance of each of the model inputs. The aim of the sensitivity analysis was to delete those inputs that do not have a significant effect on model performance. The software package used enables such sensitivity analysis to be carried out. Plots of the sensitivity of the inputs can then be inspected to determine which of the inputs can be deleted. No fixed level was used to distinguish between significant and non-significant inputs. The sensitivities were used as a guide to decide which inputs to retain and which to delete by applying some degree of judgement.

The systematic approach used in experiment #2 to identify both the optimum input and hidden layer structures were as follows;

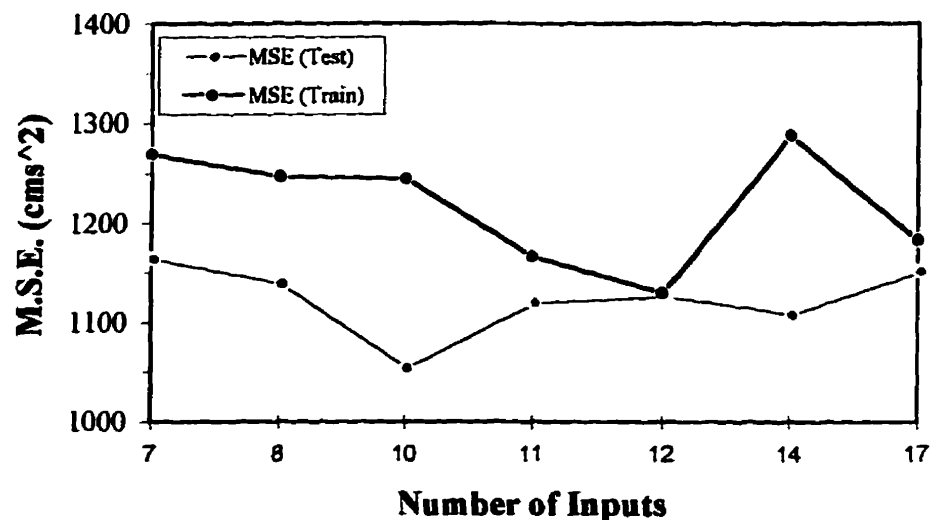
**Step 1:** For the one-week ahead forecast, start with the original 18 inputs as in experiment #1 and add the two new inputs  $Period(t+1)$  and  $Cprecip(t)$  for a total of 20 inputs. Again there is only one output from the network,  $Flow(t+1)$ . Therefore, the input layer structure included 20 nodes and the output layer structure included one node.

**Step 2:** Once the input/output structure was determined, the same systematic approach that was used in experiment #1 was followed to determine the number of nodes in the single hidden layer.

**Step 3:** Record the lowest MSE (test).

**Step 4:** Using the “Leave Out Analysis” available in the Braincel™ software, perform a sensitivity analysis on the input data and eliminate input variables that are not being used to learn the underlying process being modelled. Repeat steps 1 through 4.

**Step 5:** Plot the number of inputs versus both the MSE(test) and MSE (train) as shown in Figure 4.7.



**Figure 4.7.** Experiment #2.2, Variation of MSE for (1) training set and (2) testing set, with the number of inputs - for the one-week ahead forecast.

The network with the lowest MSE(test) is termed the optimum network because it is the one that performs the best at generalizing on new data with fewer inputs than were used in experiment #1. It was found that there was no special advantage in taking into account more than, at most, the data of the previous four periods from the week being forecasted. The optimum network consisted of 10 inputs variables in the input layer.

For the one-week ahead forecast of local inflow,  $Flow(t+1)$ , the ANN model consisted of the following 10 input variables:

- “period of the year” of the one-week ahead forecast, *Period (t+1)*
- past three periods of weekly precipitation, *P(t), P(t-1), P(t-2)*.
- cumulative precipitation since November 1<sup>st</sup> to the period of the forecast, *Cprecip(t)*.
- past week of average temperature, *T(t)*.
- past four periods of average weekly local inflow, *Flow(t), Flow(t-1), ..., Flow(t-3)*.

Both the “One-lag” and “Multi-lag” output predictions for the test samples are done with the given model structure. In the “One-lag” prediction, forecast of streamflow for each week are made based only on actual past values. The benefit of the optimum network configured in experiment #2.2 (for the one-week ahead forecast) is that the forecaster does not need to make an inference on the future weekly precipitation and average weekly temperature as was needed in experiment #1. All data used in the network for the one-week ahead forecast is historical data. In “Multi-lag” predictions, on the other hand, the 50<sup>th</sup> percentiles were used as predicted values of precipitation and temperature and previous forecasted streamflows from the neural networks were used as predicted values for streamflow. These values were appended to the database and used to predict future streamflow values. For instance, if a single prediction network is used to predict a value  $n_6$  from the observed input data  $i_1, \dots, i_5$ , then the next network prediction  $n_7$  is made using inputs  $i_2, \dots, i_5, n_6$ , and the subsequent network prediction  $n_8$  is made using inputs  $i_3, i_4, i_5, n_6, n_7$ . With the “One-lag” prediction, on the other hand, the prediction at the eighth instant is made using only the actual input data values  $i_3, i_4, i_5, i_6, i_7$ .

The network for a one-week ahead forecast consisted of 17 hidden neurons and is represented as ANN<sub>1</sub>(10, 17, 1). Similar simulations were conducted to determine the hidden node structure for the two, three and four week ahead forecasting networks and are represented as follows; ANN<sub>2</sub>(10, 5, 1), ANN<sub>3</sub>(10, 7, 1), and ANN<sub>4</sub>(10, 16, 1).

## CHAPTER 5

### 5 RESULTS AND ANALYSIS

#### 5.1 RESULTS OF FLOW FORECASTING

In this chapter, the forecasting capability of the preferred ANN, as applied to the Winnipeg River Basin, will be compared to historical values as well as results obtained from a more traditional stochastic-deterministic forecasting model for a representative range of hydrologic conditions.

##### 5.1.1 Experiment #1

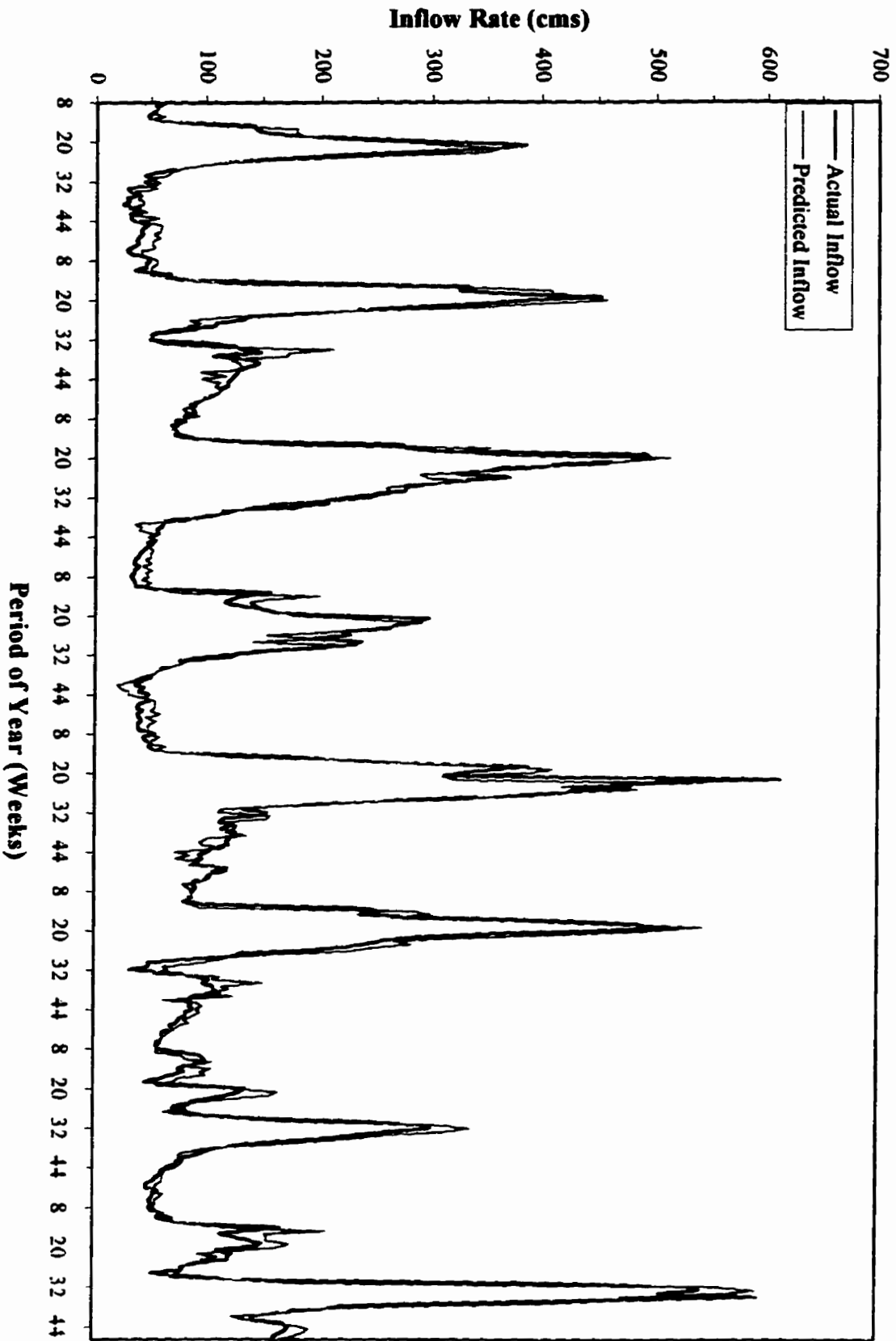
The results of forecasting using the ANN model (with the same inputs used in the WIFFS model) on the test data are presented in Figures 5.1 through 5.4 for forecast lead times of 1, 2, 3, and 4 weeks respectively. Each of these figures are of the same format. These graphs show the period of the year,  $P(t+I)$ , the forecast was made, along the x-axis with a comparison of both the historical and forecasted streamflows on the y-axis. Similar graphs can be found in Appendix B for the training data set (1965-85). The results obtained for a 1-week forecast of Namakan Lake local inflow in 1960-64 and 1986-88 are shown in Figure 5.1. The ANN seems to have had difficulty in forecasting the streamflow in period 36 of 1961. The historic precipitation and streamflow record (Appendix A) shows approximately 140 mm of precipitation in period 34 and approximately 150 cms of streamflow in period 36, both in the year of 1961. This forecasting error is probably due to one of two types of problems in the data for that time period. The first possible problem with the data is that the streamflow gauging station could have been misread, indicating a lower streamflow than what actually occurred. The second possible problem with the data is that the precipitation gauge could have been misread, indicating a higher precipitation amount than what actually occurred. Furthermore, the forecast error for period 36 of 1961 gets larger as the lead-time increases. This is expected because



forecasts of larger lead-times rely heavily on inferences of future temperature, precipitation and streamflow values.

The ANN model forecasts the magnitude and timing of both the summer peaks and the smaller peaks which occur early and late in the year, quite well. Such peak flows were detected in the 20 years used for training from 1965-85, so it is not surprising that the model did well at forecasting them in 1960-64 and 1986-88. The results obtained for the 2, 3, and 4 week forecast of Namakan Lake local inflow in 1960-64 and 1986-88, are shown in Figures 5.2-5.4. The ANN model forecasts the magnitude of the baseflow quite well, but encounters some difficulty in forecasting the magnitude of the peak flows. Generally, as the forecast lead-time grows from 2 to 4 weeks so does the forecast error. This is expected since these forecasts are beginning to use inferences on the precipitation, temperature and forecasted flow from the previous week(s).

The traditional scatter plots showing observed (historical) flows on the x-axis against the forecasted flows from the ANN on the y-axis are displayed in Figures 5.5-8 for the test set data. Similar graphs can be found in Appendix D for the training data set. In each of the scatter plots a perfect forecast lies on the 45° line. Generally, the order of the figures follows decreasing ANN performance. This is expected since forecasts with lead-times of greater than a week rely heavily on forecasts of both meteorological conditions and streamflows. Consequently, the practical limit to forecast lead-time is a function of the basin lag-time and the accuracy and availability of weather forecasts [Acres, 1993]. The current technology cannot provide forecasts of meteorological conditions beyond 3 to 5 days into the future and for a basin like Namakan Lake with a 7-day lag time, the forecast lead-time would be limited to about 10 to 12 days. Forecasts with lead times beyond 10 or 12 days are largely probabilistic. This is why the one-week ahead forecasts are fairly accurate while the two, three and four-week ahead forecasts generally worsen as the lead-time increases.



**Figure 5.1.** Best 1-week forecast of local inflow into Namakan Lake for 1960-64 & 1986-88 using the Exp. #1 ANN model.

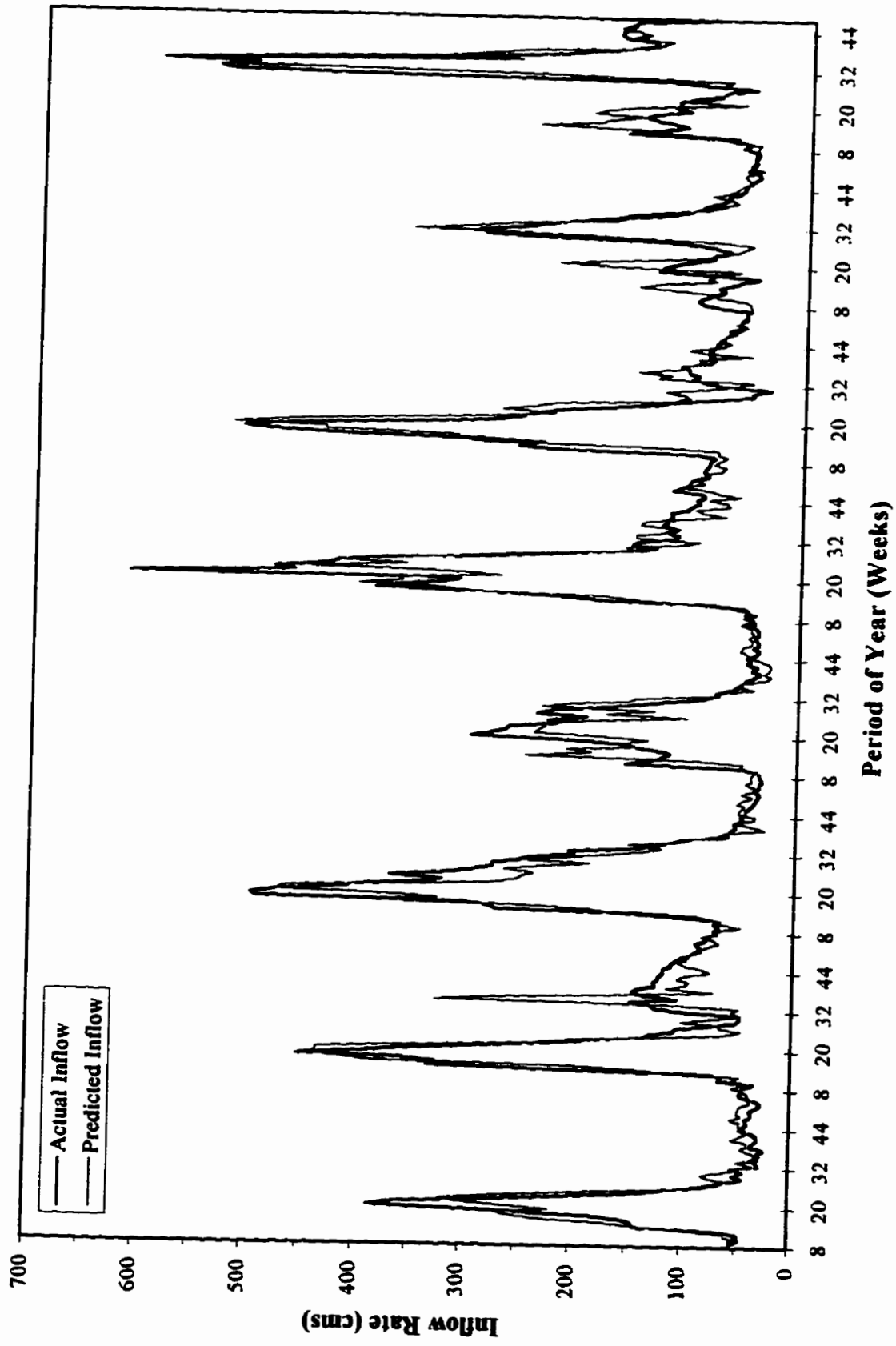


Figure 5.2. Best 2-week forecast of local inflow into Namakan Lake for 1960-64 & 1986-88 using the Exp. #1 ANN model.

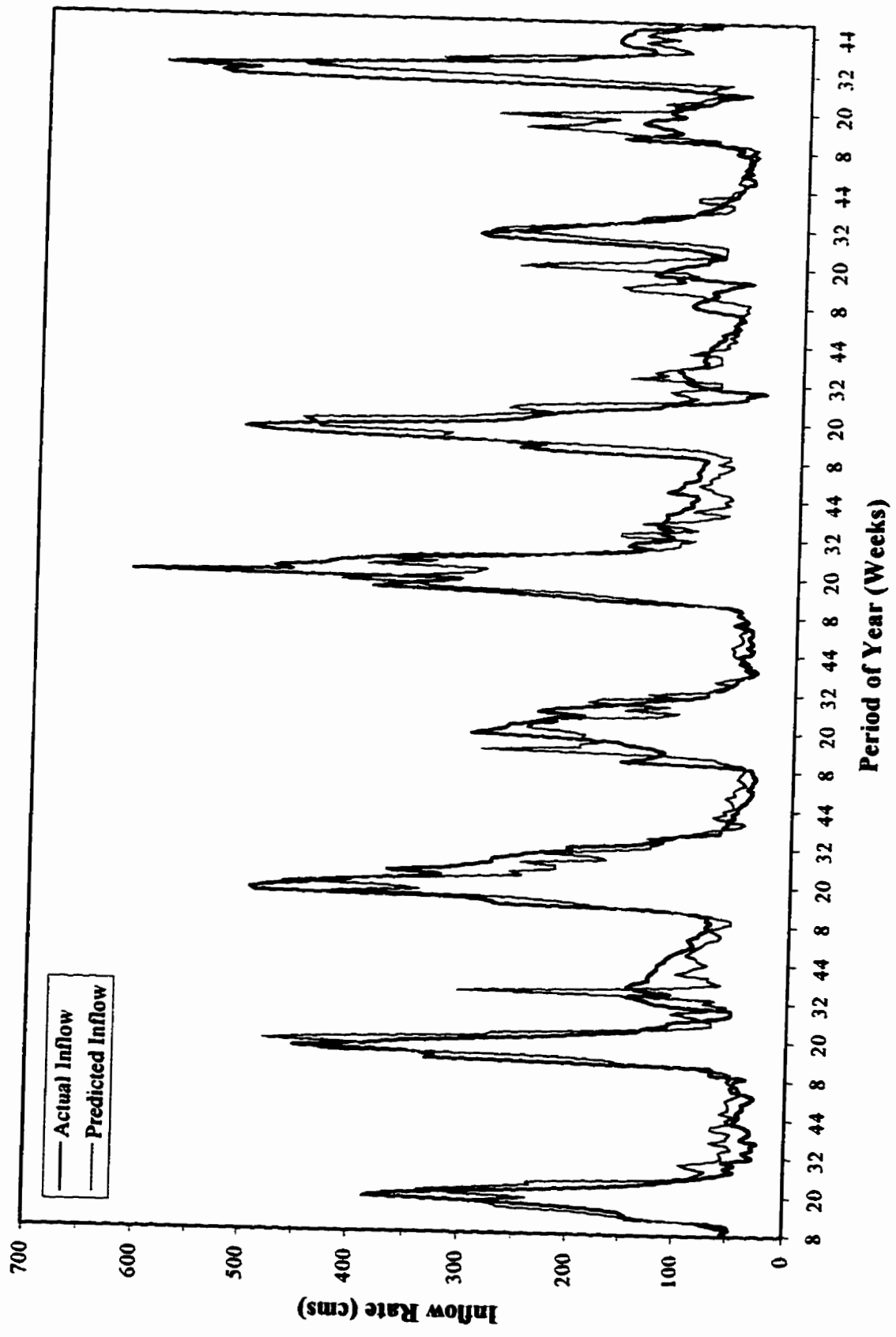


Figure 5.3. Best 3-week forecast of local inflow into Namakan Lake for 1960-64 & 1986-88 using the Exp. #1 ANN model.

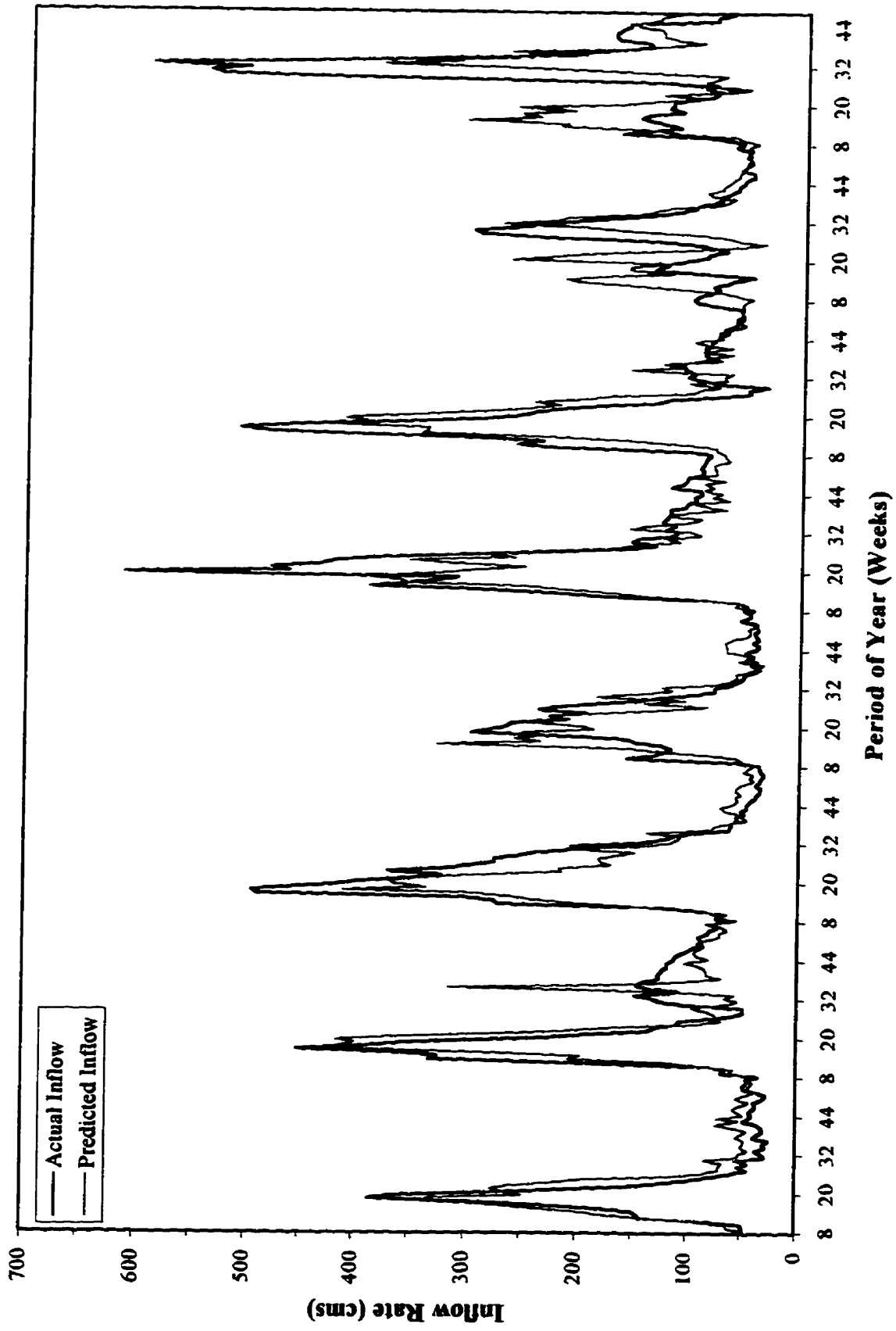
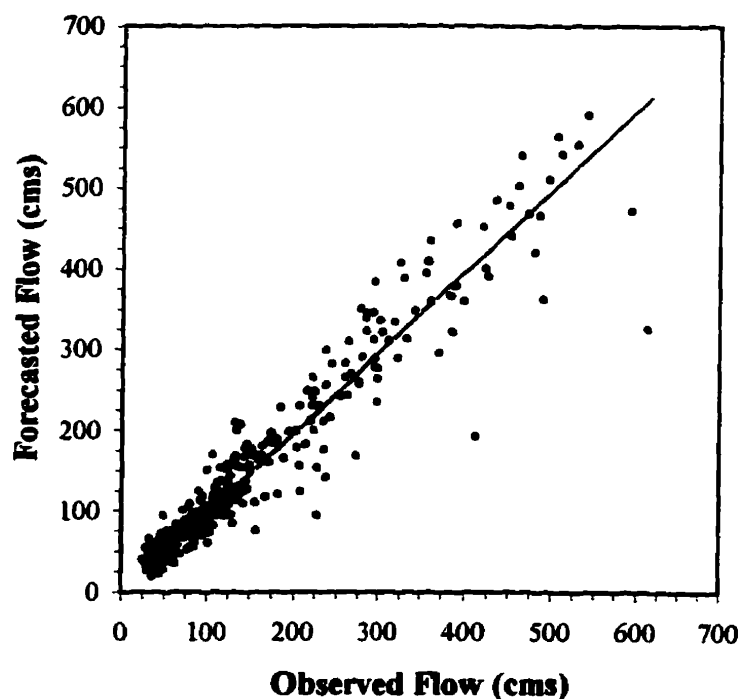
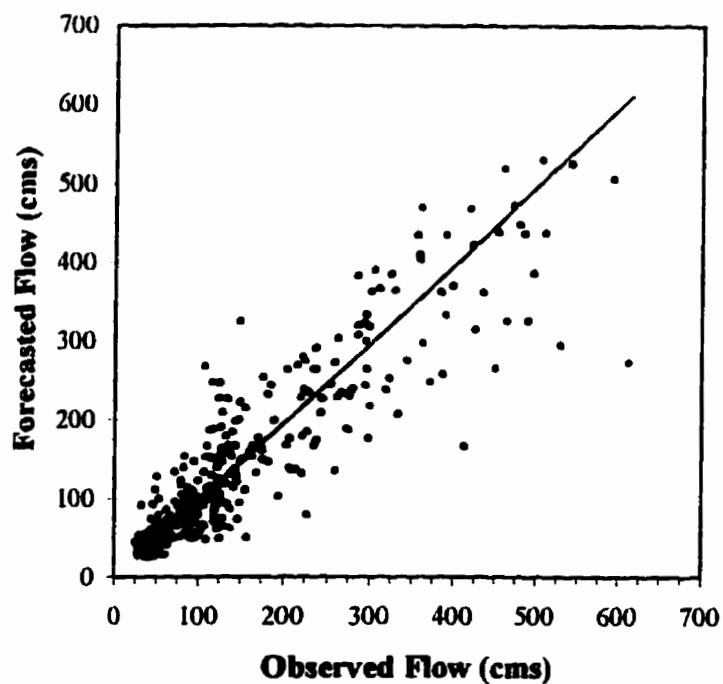


Figure 5.4. Best 4-week forecast of local inflow into Namakan Lake for 1960-64 & 1986-88 using the Exp. #1 ANN model.

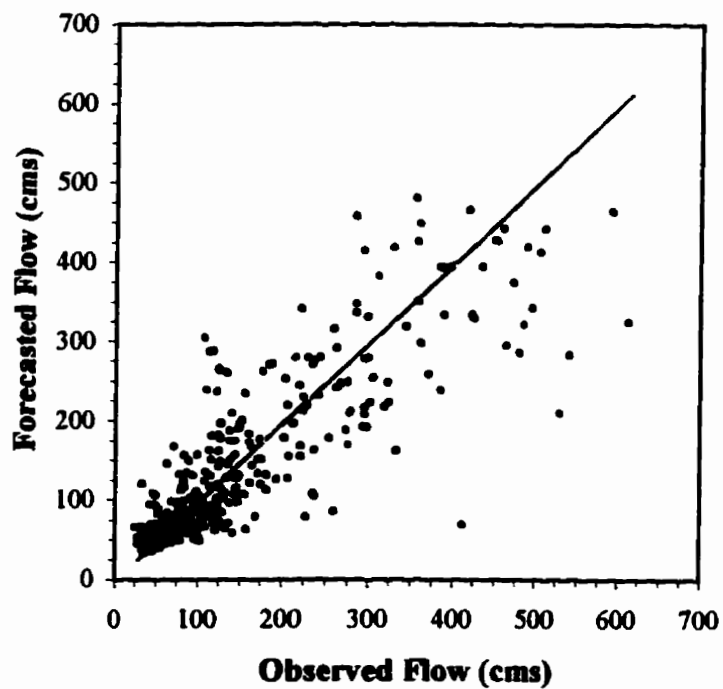
The one-week ahead forecast (Figure 5.5) falls relatively close to the 45° line except for three points. These three points are most likely problems with the input data (i.e; streamflow or precipitation gauging errors). It is interesting to note that the flows below 200 cms tend to fall closer to the 45°. This illustrates that the ANN is very accurate at forecasting baseflow values. As the flows increase above 200 cms, the forecasts tend to diverge from the 45° line showing the difficulty the ANN has with larger, peakflow values. As the lead-time increases from 1 through to 4 weeks, the forecasted flows tend to increasingly diverge from the 45° line, especially for flows greater than 200 cms. This can be seen by comparing the plots in Figures 5.6-8.



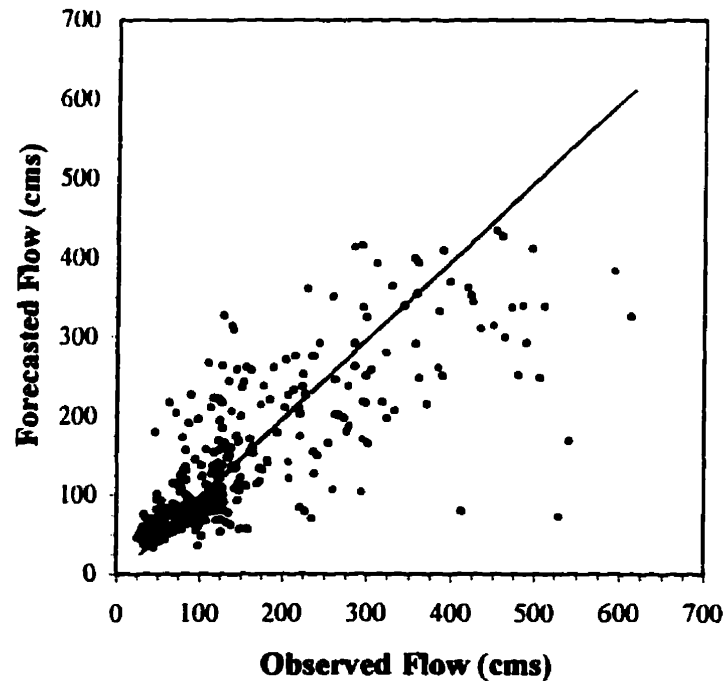
**Figure 5.5.** Observed versus Best 1-week Forecast of local inflow to Namakan Lake for 1960-64 & 1986-88 using Exp. #1 ANN.



**Figure 5.6.** Observed versus Best 2-week Forecast of local inflow to Namakan Lake for 1960-64 & 1986-88 using Exp. #1 ANN.



**Figure 5.7.** Observed versus Best 3-week Forecast of local inflow to Namakan Lake for 1960-64 & 1986-88 using Exp. #1 ANN.

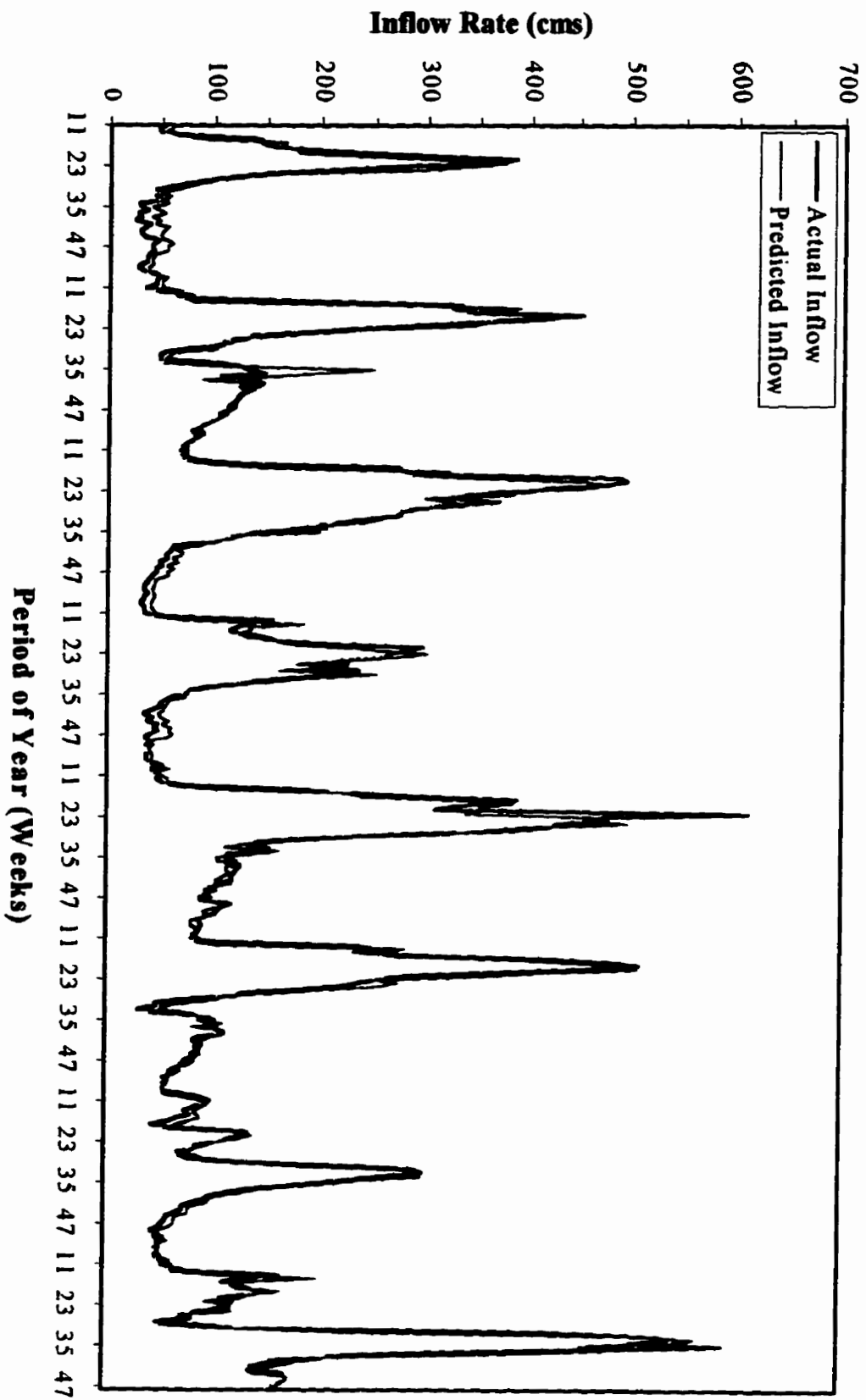


**Figure 5.8.** Observed versus Best 4-week Forecast of local inflow to Namakan Lake for 1960-64 & 1986-88 using Exp. #1 ANN.

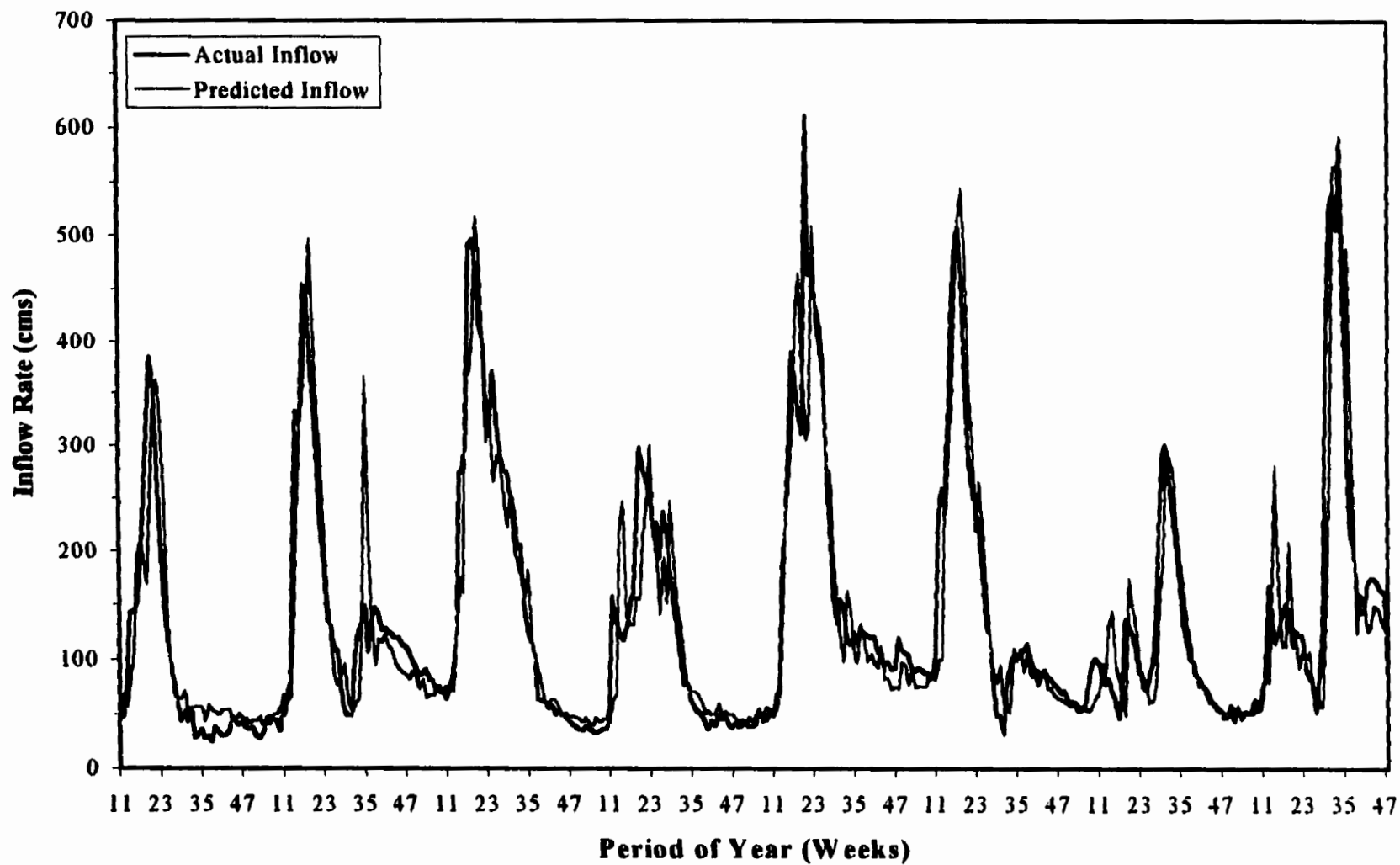
### 5.1.2 Experiment #2

Again, in the second experiment (#2.2), the ANN built was not restricted to the input variables used in the WIFFS model and the data set had part of the data (from period 33 of 1983 to period 33 of 1984) eliminated. The results of experiment #2.2 using the ANN on the test data are presented in Figures 5.9 through 5.12 for forecast lead-times of 1, 2, 3, and 4 weeks respectively. Each of the Figures 5.9-12 are of the same format. These graphs show the period of the year,  $P(t+1)$ , the forecast was made, along the x-axis with a comparison of both the historical and forecasted streamflows on the y-axis. Similar graphs can be found in Appendix C for the training data set. The traditional scatter plots showing observed (historical) flows on the x-axis against the forecasted flows from the ANN on the y-axis are displayed in Figures 5.13-16 for the test set data. Similar graphs can be found in Appendix D for the training data set. In each of the graphs in Figures 5.13-16, a perfect forecast lies on the  $45^\circ$  line. Generally the order of figures follows decreasing ANN performance. The following section quantitatively evaluates the data in each of the figures in terms of the four performance measures discussed earlier.

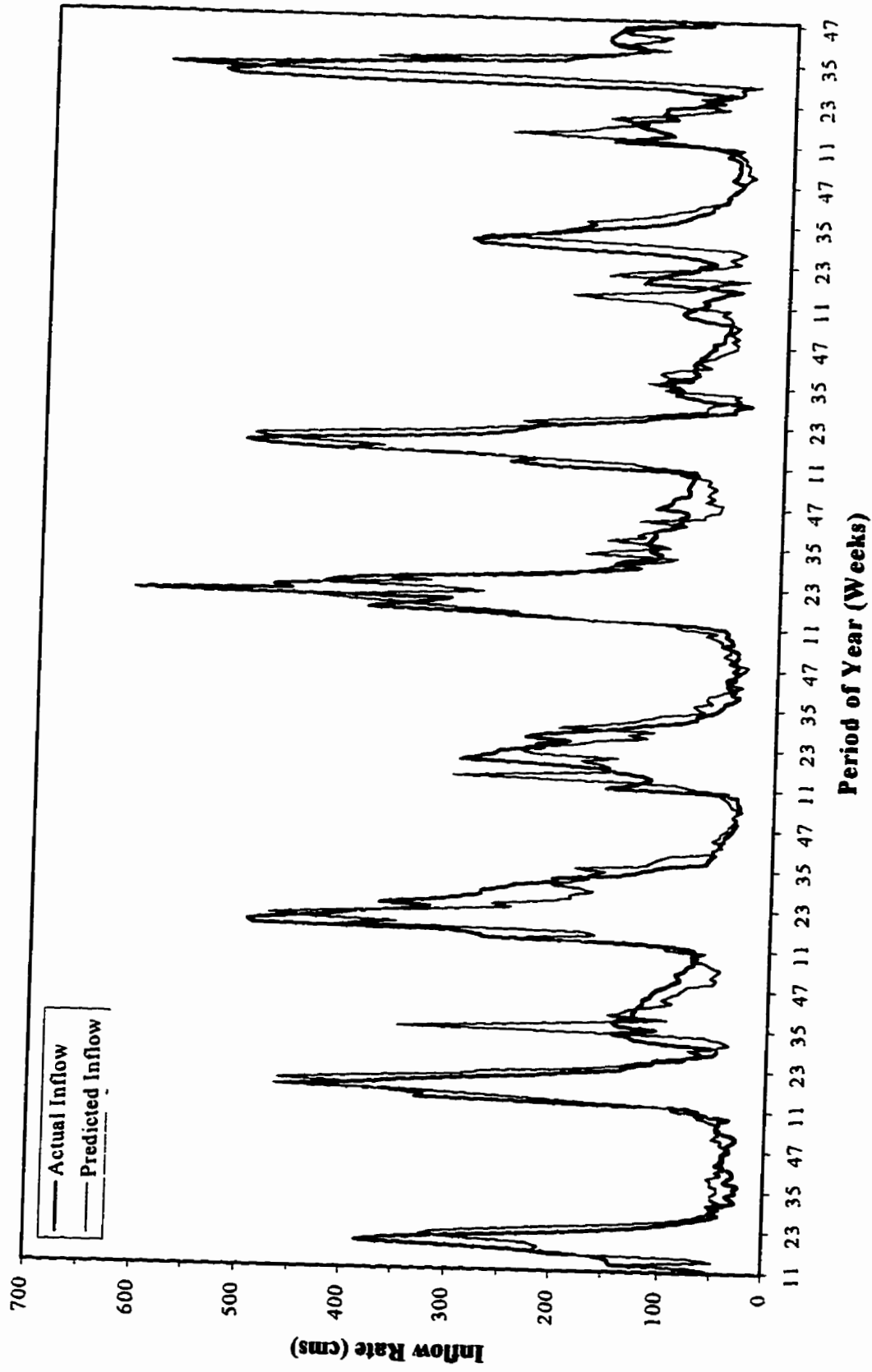




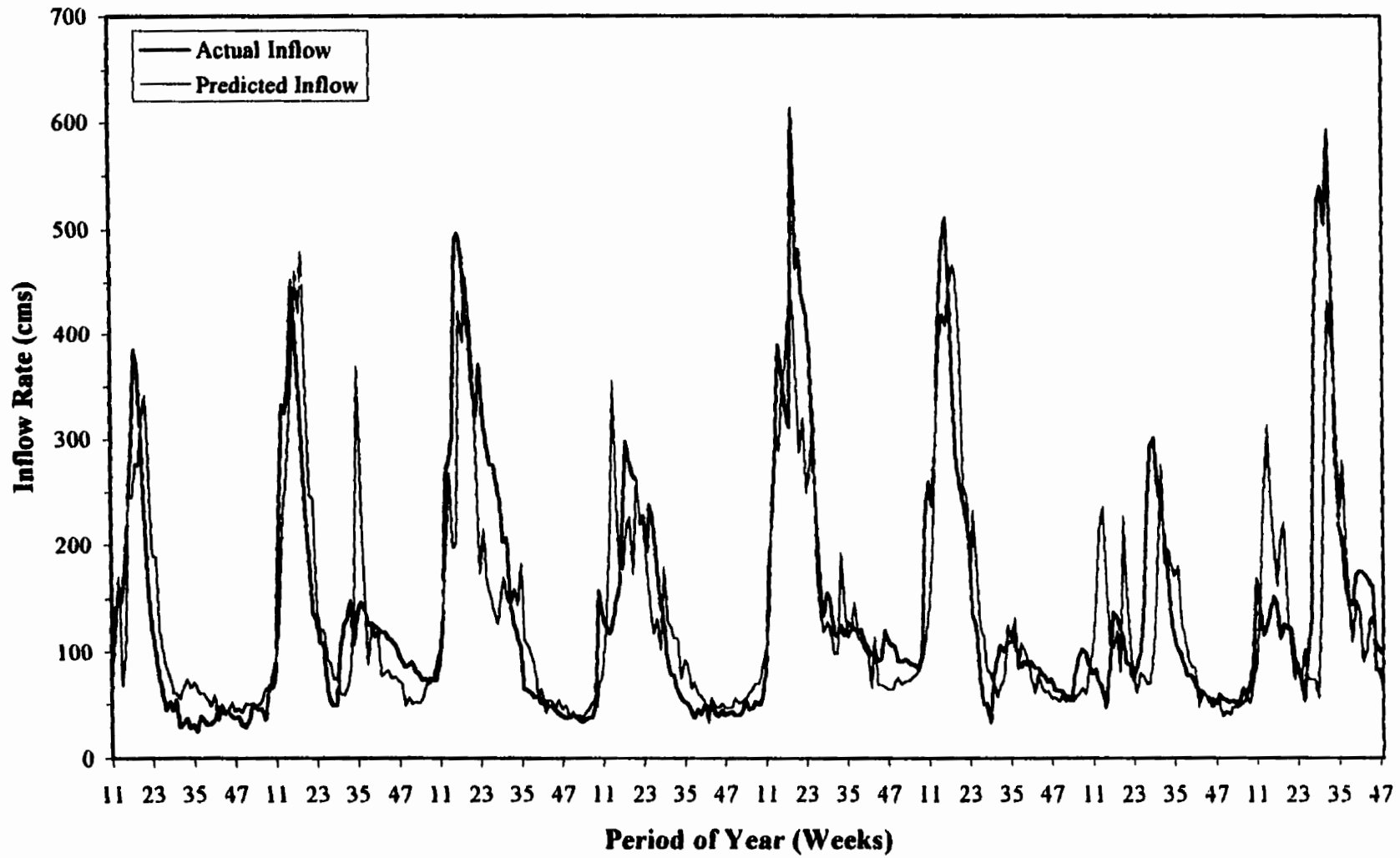
**Figure 5.9.** Best 1-week forecast of local inflow into Namakan Lake for 1960-64 & 1986-88 using the Exp. #2 ANN model.



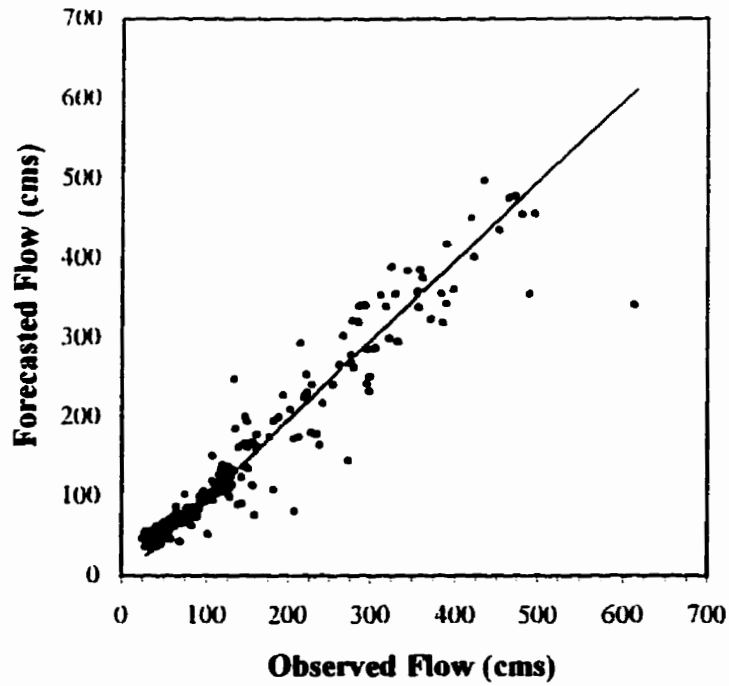
**Figure 5.10.** Best 2-week forecast of local inflow into Namakan Lake for 1960-64 & 1986-88 using the Exp. #2 ANN model.



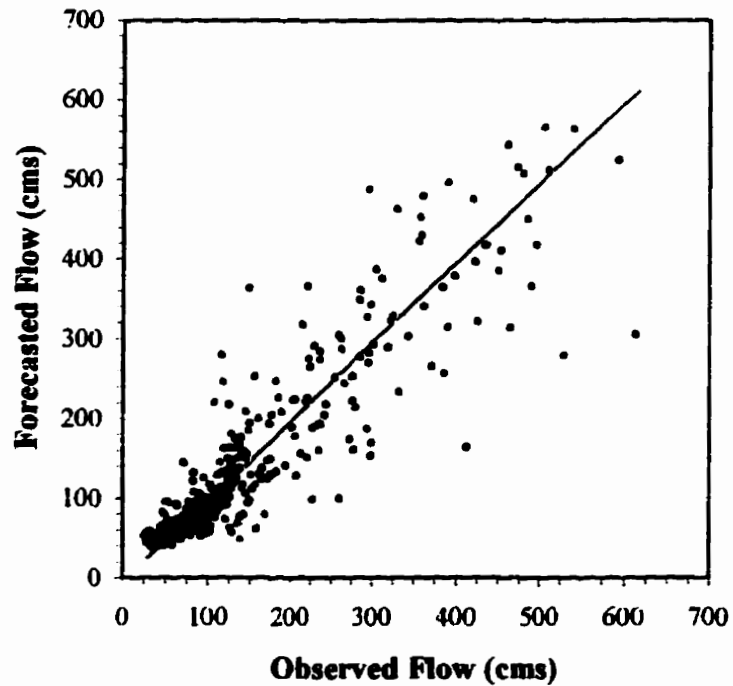
**Figure 5.11.** Best 3-week forecast of local inflow into Namakan Lake for 1960-64 & 1986-88 using the Exp. #2 ANN model.



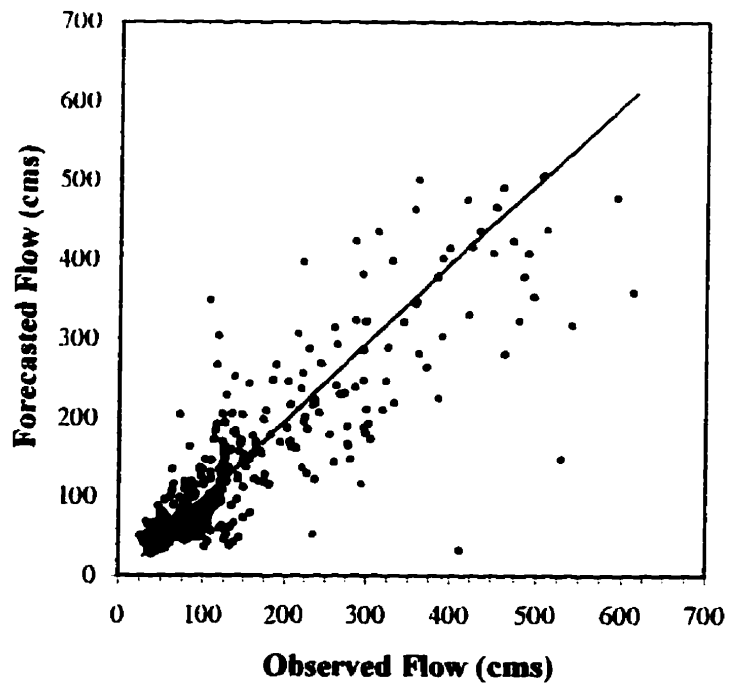
**Figure 5.12.** Best 4-week forecast of local inflow into Namakan Lake for 1960-64 & 1986-88 using the Exp. #2 ANN model.



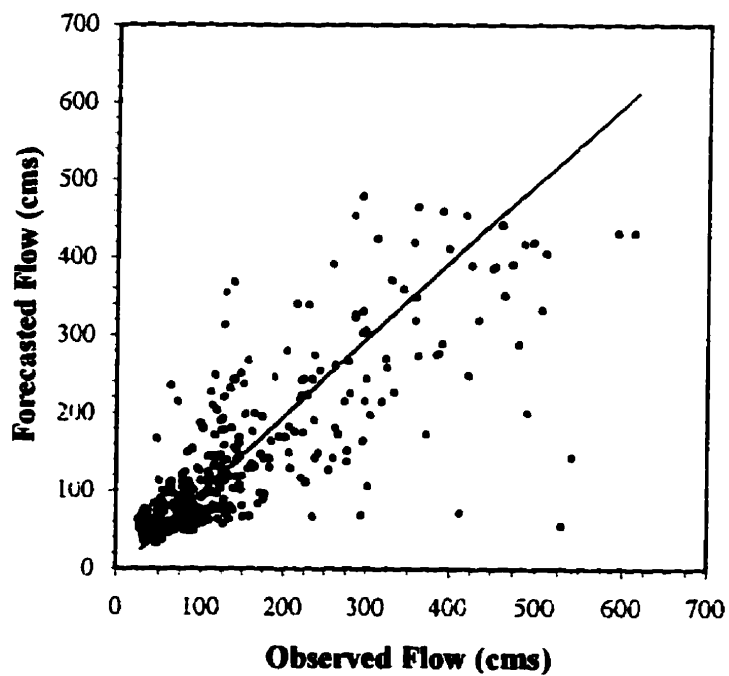
**Figure 5.13.** Observed versus Best 1-week Forecast of local inflow to Namakan Lake for 1960-64 & 1986-88 using Exp. #2 ANN.



**Figure 5.14.** Observed versus Best 2-week Forecast of local inflow to Namakan Lake for 1960-64 & 1986-88 using Exp. #2 ANN.



**Figure 5.15.** Observed versus Best 3-week Forecast of local inflow to Namakan Lake for 1960-64 & 1986-88 using Exp. #2 ANN.



**Figure 5.16.** Observed versus Best 4-week Forecast of local inflow to Namakan Lake for 1960-64 & 1986-88 using Exp. #2 ANN.

## **5.2 COMPARISON OF RESULTS**

In order to evaluate the results of forecasting the natural inflow into Namakan Lake utilizing ANNs, an alternative forecasting model was employed for comparison purposes. This model, as previously mentioned, was the WIFFS model developed by *Acres International Ltd.*, [1993]

In order to quantify the relative merit of the two forecasting techniques, it was important to determine appropriate mathematical means of measuring the effectiveness of each method. The forecasting results were based on three evaluation measures. These performance measures were computed separately for the ANN forecasts of the training data and the independent events or testing data. These measures indicate how well the ANN learned the events it was trained to recognize, and the degree to which each ANN can generalize its training to forecast events not included in the training process. The primary performance measure that compared the WIFFS model to the ANN model was i) root-mean-squared error.

The three secondary performance measures that were used (only in experiment #2) to compare the best ANN model against historical values were ii) forecast of total volume, iii) forecast of peak flow magnitude, and iv) forecast of peak flow location. These four performance measures are described below. Each of the three effectiveness measures were calculated for the calibration set (1965-85) and each of the two verification sets (1960-64 and 1986-88).

i) The root mean squared error (RMSE) measures the residual variance; the optimal value is 0.0. The RMSE of the forecast results was calculated by first squaring the difference between the measured and forecasted values. The squared values were then summed over all the forecasts in a given data set and divided by the number of forecasted values in that data set. The square root of this final number was taken to produce the RMSE.

- ii) The Percent Volume Error (% VE) statistic measures the percentage of the observed volume that is captured under the predicted hydrographs, summed over the test period; the best value is 100. The ability to forecast the total yearly volume of flow was represented by the fraction of the measured yearly volume that was forecasted by the ANN model. This value was calculated by taking the difference between the forecasted and actual volume (Forecasted Value - Actual Value) dividing this difference by the actual volume and adding this number to 100. This quantity will provide a utility measure of the ANN forecasting technique for making predictions and could be used in determining the optimal filling procedure to be pursued for Namakan Lake.
- iii) The Percent Maximum Flow Magnitude (% MFM) statistic measures the percent error in matching the maximum (peak) flow magnitude of the test set. The ability of the ANN model to forecast the magnitude of the peak flow was measured by calculating the difference between the forecasted and actual peak flow (Forecasted Value - Actual Value) and dividing this difference by the actual peak flow. The resulting quantity, multiplied by 100, is the percent error in forecasting the peak flow. The peak flow was defined as the maximum flow in the year (i.e. one per year), for each of the eight seasonal hydrographs in the test set. The best value is 0.0, indicating a perfect forecast of peak flow magnitude. Positive percentages result when the ANN forecast is greater than the actual value indicating an overestimation of the peak flow. Negative percentages result when the ANN forecast is less than the actual peak value indicating an underestimation of the peak flow.
- iv) The Location of the Maximum Flow Magnitude (LMFM) statistic measures the error in matching the location of the maximum (peak) flow magnitude of the test set. The ability of the ANN model to forecast the location of the peak flow was measured by calculating the difference between the forecasted peak location and the actual peak location (Forecasted Peak Location - Actual Peak Location), where location refers to the position on the time axis. Again, the best value is 0.0, indicating a perfect forecast of the location of the peak flow. A positive number indicated the location of the



forecasted peak lags behind the actual peak. A negative number indicates the location of the forecasted peak is located before the actual peak.

The error in predicting the magnitude and location of the peak flow is a useful means of determining the utility of the forecasting model for providing early warning of potential flood events.

### 5.2.1 Experiment #1

The ANN was able to train to a smaller RMSE than the WIFFS model in all of the four forecast leads. The improvements in RMSE during the twenty-one training years ranged from as little as 1 cms in the one-week lead to as much as 4.4 cms in the two-week lead. The real test of the ANN was the comparison to the WIFFS model for the testing periods from 1960-64 and 1986-88. The ANN was able to forecast to a smaller RMSE than the WIFFS model in all four forecast leads in all eight years of test data. The improvements in RMSE for the testing period from 1960-64 ranged from as little as 3.0 cms in the one-week lead to as much as 10.2 cms in the four-week lead.

**Table 5.1. Root Mean Squared Forecast Error, Experiment #1**

Period	Forecast Lead (weeks)	RMSE (m <sup>3</sup> /s)	
		WIFFS	ANN
Calibration (1965-1985)	1	36.5	35.5
	2	56.3	51.9
	3	69.7	68.6
	4	82.3	78.4
Verification (1960-1964)	1	36.9	33.9
	2	51.3	49.4
	3	60.9	54.6
	4	68.6	58.4
Verification (1986-1988)	1	50.4	34.1
	2	78.3	52.7
	3	80.1	71.7
	4	89.8	89.2

The greatest improvement was seen in the 1986-88 test years where the improvement in RMSE ranged from as little as 0.6 cms in the four-week lead to as much as 16.3 cms in the one-week lead. These results are summarized in Table 5.1 which lists the calculated root mean squared error for both models (WIFFS and ANN) and each of the three data sets (one calibration set and two verification sets).

## 5.2.2 Experiment #2

### 5.2.2.1 R.M.S.E

The calculated root-mean-squared error for both models (ANN experiment #2.1 and ANN experiment #2.2) and each of the three data sets is given in Table 5.2. Again, experiment #2.1 was the ANN built with the same inputs as WIFFS with period 33 of 1983 through period 33 of 1984 eliminated. Experiment #2.2 was the "optimal" ANN built with fewer inputs than the WIFFS model and the same data eliminated as in experiment #2.1. The "optimum" ANN was able to train to a smaller RMSE than the ANN model trained in experiment #1, in all of the four forecast leads

**Table 5.2. Root Mean Squared Forecast Error, Experiment #2**

Period	Forecast Lead (weeks)	RMSE (m <sup>3</sup> /s)	
		ANN (exp. #2.1)	ANN (exp. #2.2)
Calibration (1965-1985)	1	36.5	35.3
	2	55.1	48.4
	3	64.4	56.5
	4	75.9	68.4
Verification (1960-1964)	1	34.3	33.9
	2	47.2	49.0
	3	56.5	55.5
	4	61.0	62.2
Verification (1986-1988)	1	33.5	30.0
	2	58.8	53.7
	3	77.2	70.9
	4	94.0	86.0

The improvements in RMSE during the 21 training years ranged from as little as 1.2 cms in the one-week lead to as much as 7.9 cms in the three-week lead. The real test of the optimum ANN was the comparison to the ANN model trained in experiment #2.1, for the testing periods from 1960-64 and 1986-88. The optimum ANN was able to forecast to a smaller RMSE than the ANN model trained in experiment #2.1, in all but two of the forecast leads during the 8 years of test data. The improvements in RMSE for the testing period from 1960-64 ranged from as little as 0.4 cms in the one-week lead to as much as 1.0 cms in the three-week lead. The RMSE in this test period worsened from the ANN tested in experiment #2.1 by 1.8 cms and 1.2 cms in the 2-week and 4-week leads respectively. The greatest improvement was seen in the 1986-88 test years where the improvement in RMSE ranged from as little as 3.5 cms in the one-week lead to as much as 8.0 cms in the four-week lead. It is interesting to note that the overall RMSE for the entire eight years of test data (for the one-week lead) was considerably less than that of the WIFFS model (32.5 cms compared to 42.5 cms respectively).

#### 5.2.2.2 Forecasted Total Yearly Volume

The results for the percentage of the forecasted total yearly volume, %VE, are given in Table 5.3. The ANN model generally did a good job at forecasting the volume of flow that could be expected in a given year.

**Table 5.3. Percent of Total Yearly Volume Forecasted, Experiment #2**

Test Year	Forecasted Volume (% of Actual)			
	Lag 1	Lag 2	Lag 3	Lag 4
1960	106.1	104.9	100.3	113.4
1961	101.2	103.4	105.4	106.4
1962	96.5	92.5	86.2	81.8
1963	100.8	98.0	96.7	102.2
1964	96.3	95.4	97.8	90.9
1986	100.5	98.3	94.5	97.6
1987	101.0	96.4	92.7	98.4
1988	98.7	94.5	84.9	82.4
<b>Average</b>	<b>100.1</b>	<b>97.9</b>	<b>94.8</b>	<b>96.6</b>

For a lead-time of one week, the forecasted volumes ranged from an overprediction of 6.1 % to an underprediction of 3.7%. The average overprediction was 0.1%. For a lead-time of two weeks, the forecasted volumes ranged from a 4.9% overprediction to a 7.5% underprediction with an average underprediction of 2.1%. Three-week lead times produced forecasted volumes ranging from a 5.4% overprediction to a 15.1% underprediction with an average underprediction of 5.2%. Four-week lead times produced forecasted volumes ranging from a 13.4% overprediction to a 18.2% underprediction with an average underprediction of 3.4%. As expected, there was a considerable range of observed errors in forecasting the total yearly volume within both the eight years of test data as well as the four different lead times. It was encouraging though, to see that the average total forecasted yearly volume only worsened by 5.3% from a one-week ahead forecast to a three-week ahead forecast with forecasts made four weeks ahead improving by 1.8% over the three-week ahead forecast.

### 5.2.2.3 Forecasted Peak Flow Magnitude

The results for the percent error in forecasting the maximum (peak) flow magnitude, %MFM, are shown in Table 5.4. The ANN model performed well at forecasting both the magnitude and location of the eight peak flows in the test set. However, there is a considerable range of observed errors in forecasting the magnitude of the peak flow.

**Table 5.4. Percent Error in Forecast of Peak Flow Magnitude, Experiment #2**

Test Year	% Error in the Magnitude of Peak Flow			
	Lag 1	Lag 2	Lag 3	Lag 4
1960	-3.1	-6.4	-16.3	-11.7
1961	-4.1	9.6	3.0	5.7
1962	-3.9	4.1	-4.1	-8.4
1963	0.9	0.5	-19.5	-12.1
1964	-18.8	-17.0	-28.7	-29.6
1986	-0.5	6.4	-1.8	-8.8
1987	-1.0	-2.9	-2.9	-8.9
1988	-4.8	-4.6	-14.6	-27.2
<b>Average</b>	<b>-4.4</b>	<b>-1.3</b>	<b>-10.6</b>	<b>-12.6</b>

For a lead-time of one week, the forecasted peak flows ranged from an overprediction of 0.9% to an underprediction of 18.8% with an average underprediction of 4.4%. For a lead-time of two weeks, the forecasted peak flows ranged from an overprediction of 9.6% to an underprediction of 17% with an average underprediction of 1.3%. Three-week lead times produced forecasted peak flows ranging from a 3% overprediction to a 28.7% underprediction with an average underprediction of 10.6%. Four-week lead times produced forecasted peak flows ranging from a 5.7% overprediction to a 29.6% underprediction with an average underprediction of 12.6%. The year 1964 seemed to be largely underpredicted for all lead times. This is perhaps due to one or perhaps both of the following reasons. The peak flow in 1964 was approximately 612 cms which was almost as large as the extreme flow of approximately 668 cms in 1977 used in the training of the network. When the network scales the data the extreme low and high events are scaled very close to zero and one respectively. When fed through the transfer function in the hidden and output nodes, these values fall on the most linear portion of the curve and therefore are not entirely being modelled in a nonlinear fashion. This is why the results show that the ANN has a difficult time predicting the extreme high (e.g. peak of 1964) and low events (e.g. periods 35-48 of 1960) in the test set. The second reason for the underestimation of the 1964 peak flow could be that the 1963 peak flow was considerably lower at approximately 298 cms. Peak flows for the first three years (1960-62) of the test set were in the commonly higher flow ranges of approximately 400-500 cms. Then all of a sudden the next year (1963) drops to below 300 cms and the year after that (1964) jumps up past the 600 cms flow level. The ANN seems to have some trouble jumping from one extreme event to another extreme event the following year. The ANN did not see this type of situation in the training set and therefore cannot be expected to predict this situation in the testing set.

#### ***5.2.2.4 Forecasted Location of Peak Flow***

The results for the error in the location of the forecasted maximum (peak) flow magnitude, LMF<sub>M</sub>, are shown in Table 5.5. Peak flow locations for a one-week ahead forecast ranged from being two weeks ahead to three weeks behind the actual location with an

average of 0.90 weeks behind. Two and three-week ahead forecasts of peak location produced values in the range of one week ahead to three-weeks behind the actual peak. The two-week ahead forecasts averaged 1.13 weeks behind the actual peak with the three-week ahead forecast averaging 1.5 weeks behind. The four-week ahead forecast produced results that ranged from a perfect forecast of peak location to a forecast of three-weeks behind with an average of 1.88 weeks behind. It is interesting to note that the average location of the forecasted peak deteriorates very slowly when forecasting from one week ahead to a month ahead.

**Table 5.5. Error in Location of Forecasted Peak Flow, Experiment #2**

Test Year	Location of Peak Flow (weeks from actual peak)			
	Lag 1	Lag 2	Lag 3	Lag 4
1960	1	1	2	3
1961	0	1	0	3
1962	1	1	2	2
1963	2	3	3	3
1964	3	2	3	0
1986	1	1	2	2
1987	1	0	1	2
1988	-2	-1	-1	0
<b>Average</b>	<b>0.90</b>	<b>1.13</b>	<b>1.50</b>	<b>1.88</b>

### 5.3 SEPARATION OF DATA INTO TRAIN/TEST SETS

As mentioned earlier, when the ANN trains to the desired error tolerance using the training set, but performs poorly when applied to the test set, the network is either overtrained or data set design problems are probably present. *Smith and Eli* [1995] suggest that the data set problem usually results from either the training data set being too small, or it contains insufficient content to properly characterize the functional relationship between inputs and outputs. The following section discusses how one should break up the data set into the training and testing components to ensure optimum prediction capabilities from an ANN.

In order to investigate the effect of using different years of data for training and testing on the generalization ability of the network, the best network configuration for the one-week forecast was trained for six different data sets. Each of the six data sets used all 29 years of available data (except for the 1983-84 year of missing data) but were broken up into different years for training and testing. Table 5.6 outlines which years were used for testing and which for training.

**Table 5.6. Breakdown of Training/Testing Data Set.**

<b>Data Set #</b>	<b>Training Set</b>	<b>Test Set</b>
1	1965-67, 69-76 & 1978-85	1960-64 & 1986-88
2	1965-76 & 1978-85	1960-64, 1968, 1977, 1986-88
3	1960-64, 1970-72, 1976-88	1965-69 & 1973-75
4	1960-69, 1973-77, 1983-88	1970-72 & 1978-82
5	1960-62, 1966-88	1963-65
6	1960-71 & 1975-88	1972-74

Data Set #1 was trained and tested on the same data as the “optimum” ANN built in experiment #2.2 (referred to herein as the base case) except that the extreme low year of 1968 and the extreme high year of 1977 were eliminated from the training set. This was done to see how the network would perform on the test set when the extreme low and high events were not seen in the training set. Results indicate that both the RMSE(train) and RMSE(test) increased by 10.5 cms and 8.9 cms respectively. This suggests that including the extreme years in the training set improves the network ability to learn the other years in the training set. This is most likely because the extreme low and high years on record sets the boundaries for 0 and 1 respectively, for scaling the rest of the data in the training set. Any data in between this extreme range is then scaled to values in the range of say 0.1-0.9 or 0.2-0.8. The lower range of the scaling depends on how much smaller the extreme low event is to the next lowest event on record (in the training set). Likewise, the higher range of the scaling depends on how much larger the extreme high event is to the next highest event on record (in the training set). When these values are passed through the transfer function in the hidden and output nodes, they are on the

nonlinear portion of the curve providing more accurate forecasts. The reason the forecasts of both the extreme low and high event (during 1968 and 1977 respectively in this case) are never reached is because when scaled, the low event is scaled close to 0 and the high event is scaled close to 1. On the sigmoidal transfer function, values of 0 and 1 fall on the outer most linear parts of the curve. For the testing set, these results suggest that including the extreme year in the training set assisted the network in generalizing on new data for the same reasons as mentioned above for the training set.

The network in Data Set #2 was trained and tested on the same data as in the base case but with the extreme low year of 1968 and the extreme high year of 1977 eliminated from the training set and put in the test set. This was done to see how the network would perform at predicting extreme flow events that were not seen in the training range. Results indicate that the RMSE(train) decreased by 5.1 cms and the RMSE(test) increased by 20.4 cms. This suggests that including an extreme event in the training set might make it slightly harder for the network to learn the other training samples. More importantly, this suggests that including an extreme low and high event in the test set, that was not seen during the training period, can worsen the generalization capabilities of the ANN. This indicates that ANNs are poor at extrapolating on data in the test set that was not seen during training.

The network in Data Set #3 used eight years of data from the training set of the base case in the test set and the eight years of data that the base case was tested on was placed in the training set along with the remaining 13 years of training data from the base case. The years from 1965-69 and 1973-75, used as training years in the base case model, were used as test years in the ANN model, to determine the capabilities of the network for forecasting higher flow years. Results indicate that the RMSE(train) decreased by 7.3 cms and the RMSE(test) increased by 12.6. Again, this suggest that removing the high flow events from the training set will make it easier for the network to learn the other training samples but, by including these high flow events in the test set (flows that were never seen in the test set) makes it harder for the network to generalize on these samples.



Again, in Data Set #4, parts of the data set used to train the base case model were used to test the ANN while parts of the testing set from the base case model were used to train the ANN. The years from 1970-72 and 1978-82, used as training years in the base case model, were used as test years in the ANN model, to determine the capabilities of the network for forecasting years of lower flows. Results indicate that the RMSE(train) decreased by 7.4 cms and the RMSE(test) increased by 11.3 cms. This suggests that removing the lower flow events from the training set will make it easier for the network to learn the other training samples but, including these low flow events in the test set (flows that were never seen in the test set) makes it harder for the network to generalize on these samples.

In both the WIFFS model and the ANN models built in chapter 4, the data set was broken-up into a 70/30 - train/test split. That is, 70% (21 years) of the data was used to train the network and 30% (8 years) of the data was used to test the network's prediction capabilities. The current literature and the Braincel™ software manual both suggest approximately a 90/10 - train/test split of the data set. Therefore, the next two data sets (#5 and 6) were run to investigate if there is any advantage in training on more data and saving lesser amounts of data to test on. In Data Set #5, the three higher flow years from 1963-65 were used to test the ANN model. This was done to see how the network would perform at predicting the high peak of 1964 following the low peak of 1963 that the ANN had trouble with in the initial experiments. Results showed that the RMSE(train) decreased by 7.0 cms but the RMSE(test) increased by 16.0 cms. This is most likely due to the fact that the ANN was not having any difficulty forecasting the years from 1960-62 and 1986-88, the problem was forecasting the years from 1963-1965. It makes sense that the RMSE(train) would improve because the ANN did not have difficulty forecasting those years and therefore it should not have any trouble learning from those years. It also makes sense that the RMSE(test) got worse because the ANN initially had difficulty forecasting the years 1963-65 and now the MSE(test) statistic is only based on those three hard-to-forecast years. In Data Set #6, the three years from 1972-74 (that were similar to the flows in 1963-65) were used to test the ANN model. Results showed that the

RMSE(train) increased by 13.2 cms but the RMSE(test) decreased by 14.5. This can be explained in a similar manner to what happened in data set #5. The data that the ANN was having difficulty forecasting (1963-65) was placed in the training set and the data that the ANN learned well (1972-74) was placed in the test set to be forecasted. Therefore, the results indicate that the data the ANN had difficulty forecasting also had trouble in learning and the data that the ANN found easy to learn also had little difficulty forecasting. Note that in each of data sets #5 and #6, the training set included the extreme low and high events of 1968 and 1977 respectively.

A few points came out of this analysis of data sets. Firstly, including the extreme low and high events in the training set might increase the RMSE for the training set but it seems to be beneficial in improving the forecasts in the test set. Since the true test of a model is its ability to predict new data (i.e. generalize on data not seen in the training set) one should not be too concerned about the value of the RMSE(train) but be more concerned with the value of the RMSE(test). Therefore, it is recommended to include the extreme low and high years in the training set. One cannot expect the network to predict values from a range outside of what it has been trained on (i.e. poor extrapolation).

It does not matter what ratio (training/testing) the data is split into as long as the following three points are kept in mind; 1) the extreme low and high years on record are included in the training set, 2) the set of observations in the training set span the range of patterns for which the network will be used, and 3) the data you are testing on should be a representative range of realistic flow regimes.

The results obtained from splitting the data into 90/10 - train/test were not significantly different from the initial results obtained with the 70/30 - train/test split of the data. A review of the current literature indicates that there is no set rule available for splitting the data into a specific train/test ratio. However, the minimum split used seems to be 90/10 - train/test and the maximum split being 50/50 *Yang et al.* [1996] with the most common split being 66/33 and 80/20.

#### **5.4 SUMMARY**

**This chapter has revealed that the ANN forecasting technique utilized for this research is capable of providing reliable forecasts for the one-week ahead local inflow into Namakan Lake. Even though the estimates deteriorated for two, three and four-week ahead forecasts, which was expected, the overall results produced by the ANN models were more accurate than those produced by the WIFFS model.**

**The results of forecasting the local inflow, which included a representative range of realistic flow regimes for the eight test years, were quite consistent with actual flow values. This indicates that the ANN technique is applicable to a wide variety of flow conditions and exhibits the potential for a competitive alternative tool for the analysis of multivariate time-series.**

## CHAPTER 6

### 6 TRANSFERABILITY OF THE RESULTS

This chapter investigates the transferability of the results to another subwatershed in the Winnipeg River basin. An ANN model was built to forecast the one-week ahead streamflows for the Rainy Lake subwatershed resulting from local inflows. The same steps that were used in building the ANN for the previous subwatershed (Namakan Lake) were applied to the Rainy Lake subwatershed.

#### 6.1 ANN MODEL IDENTIFICATION

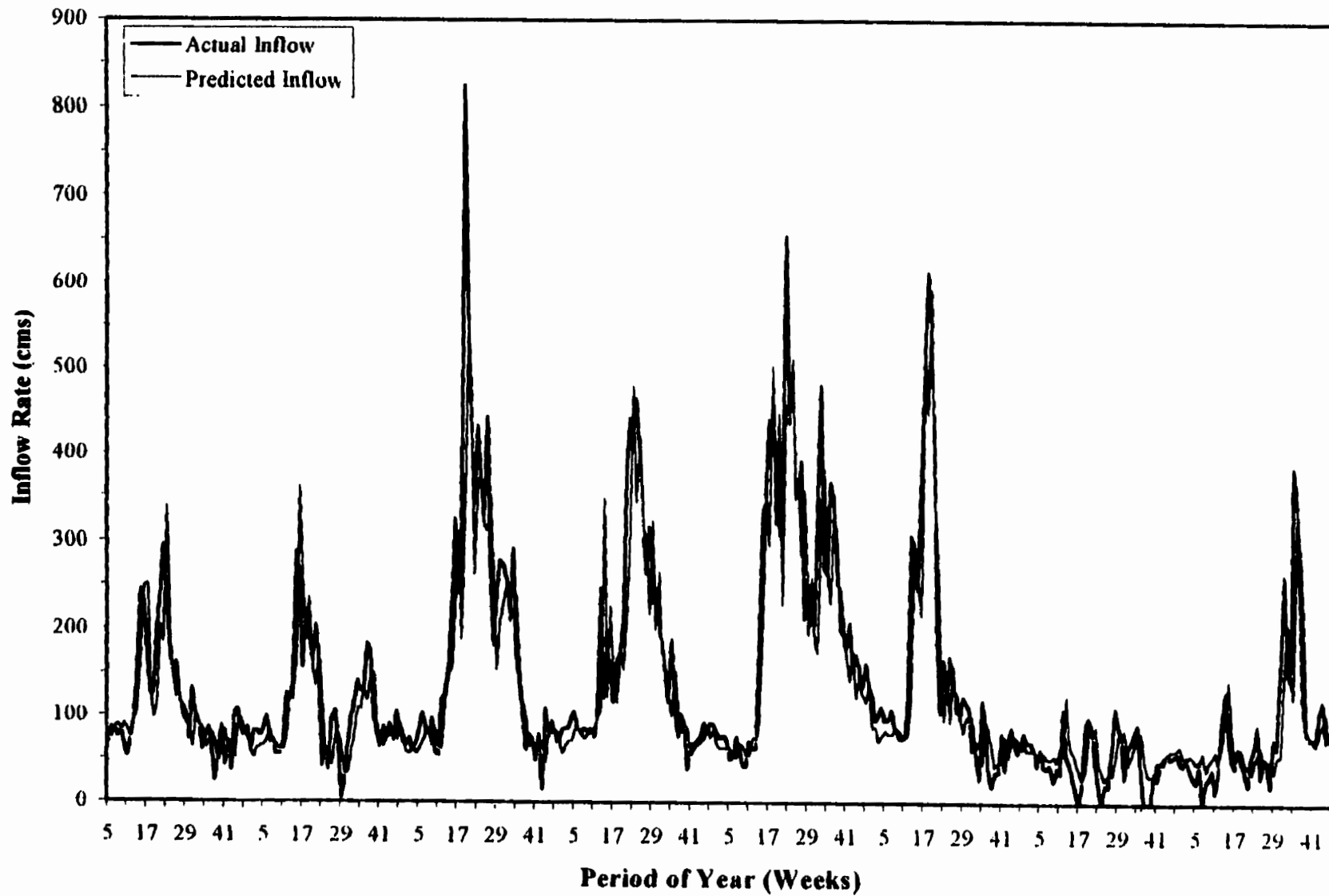
The size of the Rainy Lake subwatershed is approximately the same as the Namakan Lake subwatershed with both having approximately the same lag times of about 6 to 12 days. Therefore, the ANN model for the one-week ahead forecast of local inflow into Rainy Lake,  $Flow(t+1)$ , again consisted of the same following 10 input variables:

- “period of the year” of the one-week ahead forecast,  $Period(t+1)$
- past three periods of weekly precipitation,  $P(t)$ ,  $P(t-1)$ ,  $P(t-2)$ .
- cumulative precipitation since November 1<sup>st</sup> to the period of the forecast,  $Cprecip(t)$ .
- past week of average temperature,  $T(t)$ .
- past four periods of average weekly local inflow,  $Flow(t)$ ,  $Flow(t-1)$ , ...,  $Flow(t-3)$ .

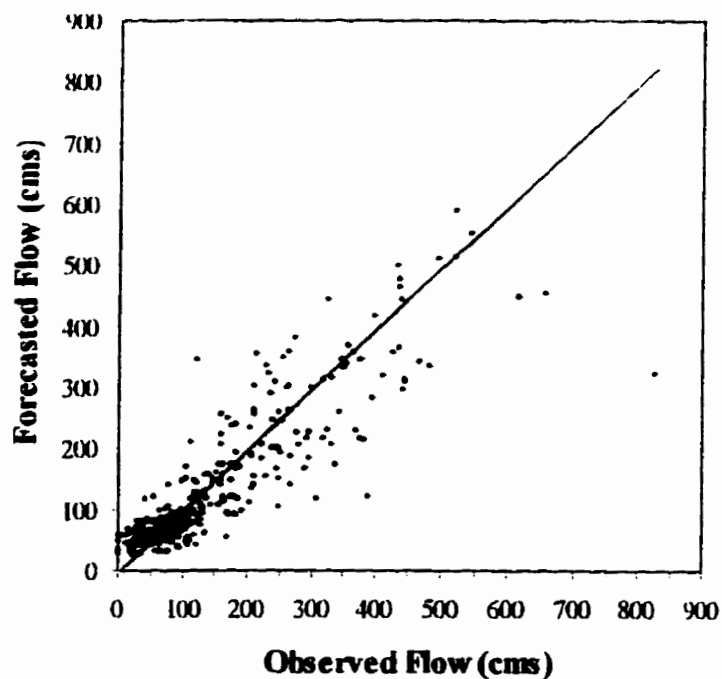
The network structure for the one-week ahead forecast consisted of a single hidden layer consisting of 5 neurons and is represented as ANN<sub>1</sub>(10, 5, 1).

#### 6.2 RESULTS OF FLOW FORECASTING

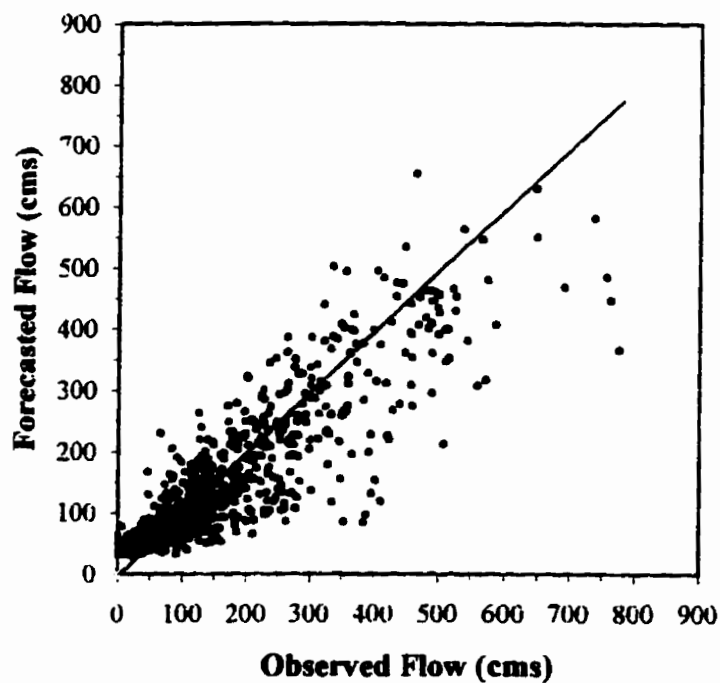
The results of forecasting using the ANN on the test data are presented in Figure 6.1 for a forecast lead-time of one-week. These graphs show the period of the year the forecast was made,  $P(t+1)$ , along the x-axis with a comparison of both the historical and



**Figure 6.1.** Best 1-week forecast of local inflow into Rainy Lake for 1960-64 & 1986-88 (test set) using the ANN model.



**Figure 6.2.** Observed versus Best 1-week Forecast of local inflow to Rainy Lake for 1960-64 & 1986-88 (test set) using the ANN.



**Figure 6.3.** Observed versus Best 1-week Forecast of local inflow to Rainy Lake for 1965-85 (train set) using the ANN.

forecasted streamflows on the y-axis. Similar graphs can be found in Appendix F for the training data set. The traditional scatter plots showing observed (historical) flows on the x-axis against the forecasted flows from the ANN on the y-axis are displayed in Figures 6.2 and 6.3 for the testing and training data sets respectively. In each of the graphs in Figures 6.2 and 6.3, a perfect forecast lies on the 45° line. It is interesting to note that the flows below 150 cms tend to fall closer to the 45°. This illustrates that the ANN is very accurate at forecasting baseflow values. As the flows increase above 150 cms, the forecasts tend to diverge from the 45° line showing the difficulty the ANN has with larger, peakflow values.

### **6.3 COMPARATIVE ANALYSIS OF WIFFS AND ANN MODELS**

#### **6.3.1 R.M.S.E**

The ANN was only able to train to a MSE that was slightly larger than the WIFFS model (3340.5 compared to 3124.8 respectively) in the one-week forecast lead. This is translated into a RMSE of 60.1 cms for the ANN and 55.9 cms for the WIFFS model as shown below in Table 6.1. The real test of the ANN was the comparison to the WIFFS model for the testing periods from 1960-64 & 1986-88. The ANN was only able to forecast to a slightly larger RMSE than the WIFFS model (63.3 cms compared to 60.6 cms respectively) in the five years of test data from 1960-64. The ANN model did a much better job than the WIFFS model at forecasting the lower baseflows during the three years for 1986-88. This is evident in comparing the RMSE of the ANN to that of the WIFFS model (47.4 cms compared to 77.8 cms respectively). These results are summarized below in Table 6.1 which lists the calculated root-mean-squared error for both models (WIFFS and ANN) and each of the three data sets (one calibration set and two verification sets) for the one-week ahead forecast.

Even though the ANN was not able to outperform the WIFFS model during the test years from 1960-64, it is interesting to note that the overall RMSE for the entire eight years of test data was considerably less than that of the WIFFS model (57.7 cms compared to 67.7 cms respectively).

**Table 6.1. Root Mean Squared Forecast Error, Rainy Lake**

Period	Forecast Lead (weeks)	RMSE (m <sup>3</sup> /s)	
		WIFFS	ANN
Calibration (1965-1985)	1	55.9	60.1
Verification (1960-1964)	1	60.6	63.3
Verification (1986-1988)	1	77.8	47.4

### 6.3.2 Forecasted Total Yearly Volume

The results for the percentage of the forecasted total yearly volume, %VE, are given in Table 6.2. The ANN model generally did a good job at forecasting the volume of flow that could be expected in a given year. For a lead-time of one-week, the forecasted volumes ranged from an overprediction of 27.0% to an underprediction of 6.0%. The average overprediction was 7.5%. The reason the overprediction was so high in 1987-88 is that during this period there were approximately eight weeks where the evaporation was greater than the inflow into the subwatershed. The data for this period was represented in the data set as negative streamflows which is physically impossible. This problem was solved by setting all negative streamflows to zero before training the network. The results in Figure 6.1 show that the network had difficulty predicting these extreme low flows causing a large overestimation in flow volume for these periods.



**Table 6.2. Percent of Total Yearly Volume Forecasted, Rainy Lake**

<b>Test Year</b>	<b>Forecasted Volume (% of Actual)</b>
	<b>Lag 1</b>
1960	103.0
1961	116.0
1962	99.0
1963	99.0
1964	94.0
1986	104.0
1987	127.0
1988	118.0
<b>Average</b>	<b>107.5</b>

### 6.3.3 Forecasted Peak Flow Magnitude

The results for the percent error in forecasting the maximum (peak) flow magnitude, %MFM, are shown in Table 6.3. The ANN model performed fairly well at forecasting both the magnitude and location of the eight peak flows in the test set. However, there is a considerable range of observed errors in forecasting the magnitude of the peak flow. For a lead-time of one week, the forecasted peak flows ranged from an overprediction of 25.3% to an underprediction of 32.8% with an average underprediction of 7.2%. The large underprediction of the peak flow in 1962 is probably because it is the highest flow on record for this subwatershed. Since this event was in the test set instead of the training set, the network had never experienced an event of such magnitude during its training. Since the network cannot predict flows outside of the range it was trained on (poor extrapolation) it is not surprising that the network could not predict the peak flow event of 1962.

**Table 6.3. Percent Error in Forecast of Peak Flow Magnitude, Rainy Lake**

<b>Test Year</b>	<b>% Error in the Magnitude of Peak Flow Lag 1</b>
1960	14.0
1961	25.3
1962	-32.8
1963	-10.2
1964	-22.2
1986	-3.7
1987	-24.3
1988	-3.5
<b>Average</b>	<b>-7.2</b>

**6.3.4 Forecasted Location of Peak Flow**

The results for the error in the location of the forecasted maximum (peak) flow magnitude, LMFM, are shown in Table 6.4. Peak flow locations for a one-week ahead forecast ranged from being exactly on time to three weeks behind the actual location with an average of 1.25 weeks behind.

**Table 6.4. Error in Location of Forecasted Peak Flow, Rainy Lake**

<b>Test Year</b>	<b>Location of Peak (weeks from actual peak) Lag 1</b>
1960	1
1961	1
1962	1
1963	1
1964	2
1986	1
1987	3
1988	0
<b>Average</b>	<b>1.25</b>

## 6.4 SUMMARY

This chapter has demonstrated the transferability of the results to another subwatershed in the Winnipeg River basin. The techniques used in chapter four to construct an ANN to forecast the local inflows into Namakan Lake were also transferable to the Rainy Lake subwatershed and are capable of providing reliable forecasts for the one-week ahead local inflow into Rainy Lake. Even though the forecasts were slightly worse for the 1960-64 test period the overall results produced by the ANN model was more accurate than those produced by the WIFFS model.

The extent to which knowledge and experience from the Namakan Lake data set was transferable to the Rainy Lake data set was in the inputs to the ANN. Since the Rainy Lake subwatershed was approximately the same size as the Namakan Lake subwatershed with approximately the same basin response times, the same number of lags were used for the precipitation, temperature and streamflow inputs. More importantly was the transferability of the period of the year at the time of the forecast, *Period (t+1)* as well as the cumulative precipitation from November 1<sup>st</sup> to the period being forecasted, up until April 1<sup>st</sup>.

If one was going to tackle another river in the Winnipeg River Basin it would be beneficial to conduct an analysis of the basin response times to obtain a good starting estimate of the number of lags of precipitation, temperature and streamflow to include as inputs to the ANN. Furthermore, if one was to apply this model elsewhere in Manitoba or other parts of Canada, it would not only be advantageous to conduct an analysis of the basin response times but it would also be advantageous to determine the physical processes involved in the resulting flow regimes. An example of knowing which physical processes translate into flow is that of the precipitation that falls during warm temperatures. In parts of Western Canada, the snow that falls during the period from November 1<sup>st</sup> to April 1<sup>st</sup> melts quickly causing immediate contributions to the streamflow. Therefore, for watersheds under these conditions, the cumulative precipitation input, *Cprecip (t)*, would not be of much use to the ANN. In the Namakan Lake subwatershed, the temperatures are cool

enough during this period to cause the snowfall to accumulate over the winter causing large runoff due to this snowmelt come spring time. In this situation, the cumulative precipitation input, *Cprecip (t)*, proved to be useful to the ANN.

## **CHAPTER 7**

### **7 CONCLUSIONS AND RECOMMENDATIONS**

The neural network is a relatively new technology and improvements are being made quite rapidly. The ANN model applied to the streamflow forecasting problem seems to have reached encouraging results for the subwatersheds under examination. A very close fit was obtained during the training phase and the networks developed consistently outperformed the WIFFS model during the testing phase. A comparison of fit between the WIFFS model and the ANN model developed in Experiment #2.2 for the Namakan Lake subwatershed revealed the following; the RMSE(test) decreased by 10 cms from 42.5 cms to 32.5 cms for the one-week lead, the RMSE(test) decreased by 12.1 cms from 62.9 cms to 50.8 cms for the two-week lead, the RMSE(test) decreased by 6.9 cms from 68.8 cms to 61.9 cms for the three-week lead, and the RMSE(test) decreased by 5 cms from 77.3 cms to 72.3 cms for the four-week lead. Furthermore, a comparison of fit between the WIFFS model and the ANN model developed for the Rainy Lake subwatershed revealed that the RMSE(test) decreased by 10 cms from 67.7 cms to 57.7 cms for the one-week lead.

The results obtained with ANNs for one, two, three, and four-week ahead forecasts are significantly better than those reached in the WIFFS model and confirm the ability of this approach to provide a useful tool in solving problems in hydrology and more specifically, streamflow forecasting. The initial success of the ANN models developed for both the Namakan Lake and Rainy Lake local subwatersheds indicated the transferability of the ANN technology to other watersheds in the Winnipeg River Basin as well as a bright future for further applications in other watershed basins in Manitoba.

Working with these simple subwatersheds, the main indications about ANN modelling features have been confirmed: i.e. their ability to self-calibrate a well balanced set of

linking weights and the possibility of being applied at the real system level. ANNs are far from being a universal tool for all computing situations. Because the ANN approach presented here does not provide models that have physically realistic components and parameters, it is by no means a substitute for conceptual watershed modelling. However, the results suggest that the ANN approach may provide a superior alternative to the time-series approach for developing input-output simulations and forecasting models in situations that do not require modelling of the internal structure of the watershed. In addition, there were a few instances where possible data problems in the case study were presented. The ANN performed poorly on these cases by significantly overpredicting the streamflows. This demonstrates that the ANN technology can also be used as an effective tool for data screening. Nevertheless, it is an efficient tool with great potential for handling natural problems where the relationships between inputs and outputs are not thoroughly explained.

The potential of ANN models for simulating the hydrologic behavior of watersheds has been presented in this research. The greatest difficulty was determining the appropriate model inputs for such a complex problem. Although ANNs belong to the class of *data-driven* approaches, it is important to determine the dominant model inputs, as this reduces the size of the network and consequently reduces the training times and increases the generalization ability of the network for a given data set. In the case study considered, sensitivity analyses were used in conjunction with judgement to reduce the number of model inputs from 18 to 10 (for the one-week forecast model), which reduced training times from approximately ten minutes to under two minutes and reduced the RMSE(test) from 33.3 cms to 32.5 cms. Similar improvements were observed for the two, three, and four-week forecast models.

It is recommended to include the extreme low and high years in the training set. One cannot expect the network to predict values from a range outside of what it has been trained on (i.e; poor extrapolation). It does not matter what ratio (training/testing) the data is split into (70/30, 90/10, 80/20 etc.) as long as more data is used to train the

network than is used to test the network. The following three points should be kept in mind when designing the training/testing data sets; 1) the extreme low and high years on record are included in the training set, 2) the set of observations in the training set span the range of patterns for which the network will be used, and 3) the testing data should be a representative range of realistic flow regimes. A review of the current literature indicated that there is no set rule available for splitting the data into a specific train/test ratio. The amount of data used for training and testing an ANN is therefore left up to the individual.

The success of an ANN implementation is dependant not just on the quality of the data used for training, but also on the type and structure of the ANN adopted, the method of training, and the way in which both the inputs and outputs are structured and interpreted. Designing a successful ANN for a specific problem doesn't require as much expertise as it does your imagination!

## **CHAPTER 8**

### **8 FUTURE WORK**

The research described in this thesis has identified potential avenues for future work. The following is a description of several additions that could be made to this research in order to increase its potential application.

It would be advantageous to explore the issue of data requirements for effective model building. ANNs are much like humans that is, the more examples given to us to learn from, the better we understand the task at hand and the easier the task becomes. Therefore, there is no limit as to how large the data set should be but is there a limit to how small the data set becomes? For both the Namakan Lake and Rainy Lake case studies there were 30 years of weekly data available for the ANNs to determine the underlying function between the inputs and the outputs. What if some of the other subwatersheds had fewer years of available data say twenty, ten or say five years? Would it be worthwhile attempting to build an ANN model for these subwatersheds with so few years of data? These are important questions that could be investigated in future research.

An addition to this research would be to build seven additional ANNs to forecast the local inflow to the each of the remaining seven subwatersheds. Furthermore, work could be implemented to network all nine subwatershed together through one ANN. That is, the ANN output from the most upstream subwatershed would serve as inputs to ANNs of downstream subwatersheds.



## REFERENCES

- Acres International Ltd (1993). "Winnipeg River Basin Flow Forecast Model Development Report", Niagara Falls, Ontario, CANADA.
- Baum, E. B., and D. Haussler (1989). "What size net gives valid generalization?". *Neural Comput.*, 1, 151-160.
- Bender M.J., Simonovic S., Booy C., and D. Cormie (1992). "Development of Long Range Flow Forecasting Capabilities for a Large Multi-Reservoir System." M.Sc. Thesis, University of Manitoba, Winnipeg, Manitoba, CANADA.
- Box, G.E.P., and G.M. Jenkins (1976). "Time Series Analysis Forecasting and Control." Holden Day, Oakland, Calif.
- Carriere, P., Mohaghegh S. and R. Gaskari (1996). "Performance of a Virtual Runoff Hydrograph System." *J. of Wat. Resour. Plan. And Mgmt.*, Nov./Dec., 421-427.
- Chakraborty, K., Mehrotra K., Mohan C.K., and S. Ranka (1992). "Forecasting the behavior of Multivariate Time Series using Neural Networks." *Neural Networks*, 5, 961-970.
- Cheng, X. and M. Noguchi (1996). "Rainfall-runoff modelling by neural network approach." *Proc. of the Int. Conf. on Water Resour. & Environ. Res. (Vol. II)*, Oct. 29-31, Kyoto, Japan.
- Dandy, G. and H. Maier (1996). "Use of Artificial Neural Networks for Real Time Forecasting of Water Quality." *Proc. Of the Int. Conf. On Water Resour. And Environ. Res. (Vol II)* Oct. 29-31, Kyoto, Japan.
- DeSilets, L., Golden B., Wang Q., and R. Kumar (1992). "Predicting salinity in the Chesapeake Bay using backpropagation." *Comput. Oper. Res.*, 19(3/4), 277-285.
- Duan, Q., Sorooshian S., and V.K. Gupta (1994). "Optimal use of SCE-UA Global Optimization method for calibrating Watershed models." *J. Hydrol.*, 158, 265-284.
- Duan, Q., Sorooshian S., and V.K. Gupta (1992). "Effective and Efficient Global Optimization for Conceptual Rainfall-Runoff Models." *Water Resour. Res.* 28(4), 1015-1031.
- Eldin, N.N. and A.B. Senouci (1995). "Condition rating of rigid pavements by neural networks." *Can. J. Civ. Eng.*, 22: 861-870.

- Flood, I. and N. Kartam (1994). "Neural networks in Civil Engineering. I: Principles and Understanding." *J. of Comp. In Civ. Eng.* 8(2), April, 131-147.
- Flood, I. and N. Kartam (1994). "Neural networks in Civil Engineering. II: Systems and Applications." *J. of Comp. In Civ. Eng.* 8(2), April, 149-162.
- Freeman J.A. and D.M. Skapura (1991). "Neural Networks: Algorithms, Applications, and Programming Techniques." Addison-Wesley Publishing Company.
- French, M.N., Krajewski W.F., and R.R. Cuykendall (1992). "Rainfall forecasting in space and time using a neural network.", *J. Hydrol.*, 137, 1-31.
- Hammerstrom, D (1993). "Working with Neural Networks", *IEEE Spectrum*, July issue.
- Hecht-Nielsen, R. (1988). "Neurocomputing: Picking the human brain." *IEEE Spectrum*, March Issue.
- Hecht-Nielsen, R. (1987). "Kolmogorov's mapping neural network existence theorem", paper presented at First IEEE International Joint Conference on Neural Networks, Inst. of Electr. and Electron. Eng., San Diego, Calif., June 21-24.
- Hertz et al. (1991). "Introduction to the Theory of Neural Computation." Lecture Notes Volume I, Santa Fe Institute, Studies in the Sciences of Complexity.
- Hipel K.W. (1986), "Time Series Analysis in Perspective.", *Water Resources Bulletin*, 21(4), 609-623.
- Hirose, Y., Yamashita K., and S. Hijya (1991). "Back-propagation algorithm which varies the number of hidden units.", *Neural Networks*, 4(1), 61-66.
- Hsu K., Gupta H.V., and S. Sorooshian (1995). "Artificial neural network modeling of the rainfall-runoff process.", *Wat. Resour. Res* 31(10), 2517-2530.
- Jayawardena, A.W. and D. Achela K. Fernando (1996). "Use of artificial neural networks in estimating evaporation." *Proc. Of the Int. Conf. On Water Resour. and Environ. Res. (Vol I) Oct 29-31, Kyoto, Japan.*
- Kachroo, R.K. (1992). "River flow forecasting. Part I: A discussion of the principles." *J. of Hydrology*, 133, pp. 1-15.
- Karnin, E.D. (1990). "A simple procedure for pruning back propagation trained neural networks.", *IEEE Trans. Neural Networks*, 1(2), 239-242.
- Karunanithi, N., Grenney, W.J., Whitley, D., and K. Bovee (1994). "Neural Networks for River Flow Prediction." *J. of Comp. in Civ. Engrg., ASCE*, 8(2), 201-220.

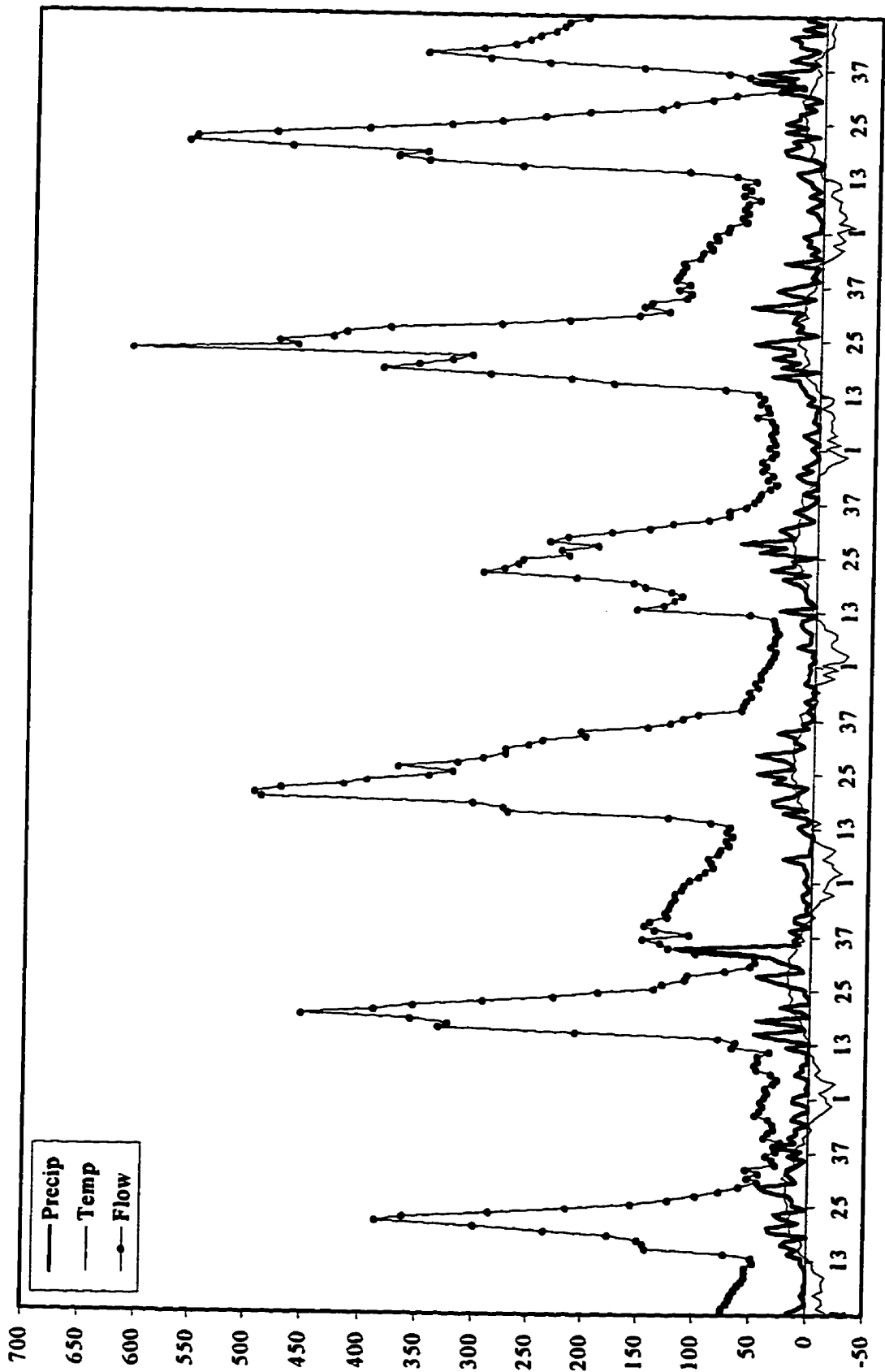
- Kitanidis, P.K, and R.L. Bras (1980a). "Adaptive Filtering Through Detection of Isolated Transient Errors in Rainfall-Runoff Models", *Water Resour. Res.*, 16(4), 740-748.
- Kitanidis, P.K, and R.L. Bras (1980b). "Real-time Forecasting with a Conceptual Hydrological Model.", *Water Resour. Res.*, 16(4), 740-748.
- Lake of the Woods Control Board (LWCB) (1994). "Winnipeg River Basin Flow Forecast Model/User Manual.", Acres International LTD., Niagara Falls, Ontario, CANADA.
- Lake of the Woods Control Board (LWCB) (1994). "Managing the Water Resources of the Winnipeg River Drainage Basin.", Ottawa, Ontario, CANADA.
- Lippmann, R.P. (1987). "An Introduction to Computing with Neural Nets.", *IEEE ASSP Magazine*: 4-22, April issue.
- Lorrai, M. and G.M. Sechi (1995). "Neural nets for modeling rainfall-runoff transformations." *Wat. Resour. Man.*, 9, 299-313.
- Maidment, D.R. (1993). *Handbook of Hydrology*. McGraw Hill, Inc., New York.
- Maier, H.R. and G.C. Dandy (1996). "The use of Artificial Neural Networks for the prediction of water quality parameters.", *Water Resour. Res.* 32(4), 1013-1022.
- Maren, A., Harston C., and R. Pap (1990). "Handbook of Neural Computing Applications.", 448 pp., Academic, San Diego, Calif.
- Masters, T (1993). "Neural Networks.", Academic Press, San Diego, Calif.
- Panu, U.S. and T.E. Unny (1980). "Extension and Application of feature prediction model for synthesis of hydrologic records." *Wat. Resour. Res.* 16(1), 77-96.
- Promised Land Technologies Inc. (1993), *Braincel User Manual, Version 2.0*, New Haven CT, United States.
- Raman H. and N. Sunilkumar (1995). "Multivariate modelling of water resources time series using artificial neural networks." *Journal of Hydrological Sciences*, 40(2), 145-163.
- Raman, H. and V. Chandramouli (1996). "Deriving a General Operating Policy for Reservoirs Using Neural Networks." *J. of Wat. Resour. Plan. And Man.*, Sept./Oct., 342-347.

- Ramesh, T.S.V. and P.P. Mujumdar (1996). "Rainfall Forecasting using Neural Networks.", Proceeding of the seventh IAHR International Symposium on Stochastic Hydraulics, July 29-31.
- Ranjithan, S., Eheart J.W. and J.H. Garrett, JR. (1993). "Neural Network-Based Screening for Groundwater Reclamation Under Uncertainty.", *Water Resour. Res.* 29(3), 563-574.
- Rodriguez, M.J and J.B. Serodes (1996). "Neural network based modeling of the adequate chlorine dosage for drinking water disinfection.", *Can J. Civ. Eng* (23), 621-631.
- Sietsma, J and R.J.F. Dow (1991). "Creating artificial neural networks that generalize.", *Neural Networks*, Vol. 14, 67-79.
- Singh, V.P. (1992), "Elementary Hydrology", Prentice-Hall, Englewood Cliffs, N.J.
- Smith, J. and R.N. Eli (1995). "Neural network models of rainfall-runoff process.", *J. Wat. Resour. Plan. and Man.*, Nov./Dec., 499-508.
- Sorooshian, S., Daun Q. and V.K. Gupta (1993). "Calibration of Rainfall-Runoff Models: Application of Global Optimization to the Sacramento Soil Moisture Accounting Model.", *Water Resour. Res.* 29(4), 1185-1194.
- Stedinger, J.R. and M.R. Taylor (1982). "Synthetic streamflow generation – model verification and validation." *Wat. Resour. Res.* 18(4), 909-918.
- Szollósi-Nagy, A. (1987) "Chapter 11 - Hydrologic forecasts and warnings." *Applied Surface Hydrology*, Water Resources Publications, 521-631.
- Tong, H. (1983). "Threshold models in non-linear time-series analysis." *Lecture Notes in Statistics*, 21, New York: Springer-Verlag.
- Tong, H. (1990). "Non-linear time series: A dynamical system approach." Oxford: Oxford University Press.
- Valencia, R.D. and Schaake, J.C. (1973). "Disaggregation processes in stochastic hydrology.", *Wat. Resour. Res.* 9(3), 580-585.
- Valluru B. R. and R.V. Hayagriva (1993). "C<sup>++</sup>, Neural Networks and Fuzzy Logic", Management Information Source, Inc., 1993.
- Wasserman, P.D. (1990). "Neural Computing: Theory and Practice", Van Nostrand Reinhold.

- Weigend, A. S., D. E. Rumelhart, and B. A. Huberman (1990). "Predicting the future: A connectionist approach", *Int. J. Neural Syst.*, 1(3), 193-209.
- Yang, C., S.O. Prasher, and R. Lacroix (1996). "Applications of Artificial Neural Networks to Simulate Water-Table Depths Under Subirrigation.", *Can. Water Resour. Journal*, 21(1), 27-44.
- Yapo, P., Gupta V.K. and S. Sorooshian (1996), "Calibration of Conceptual Rainfall-Runoff Models: Sensitivity to calibration data.", *J. Hydrol.*, 181, 23-48.

**APPENDIX A:**

**NAMAKAN LAKE HISTORICAL DATA**



Period of Year (Weeks)

Figure A-1. Namakan Lake Historical Data, 1960-65.

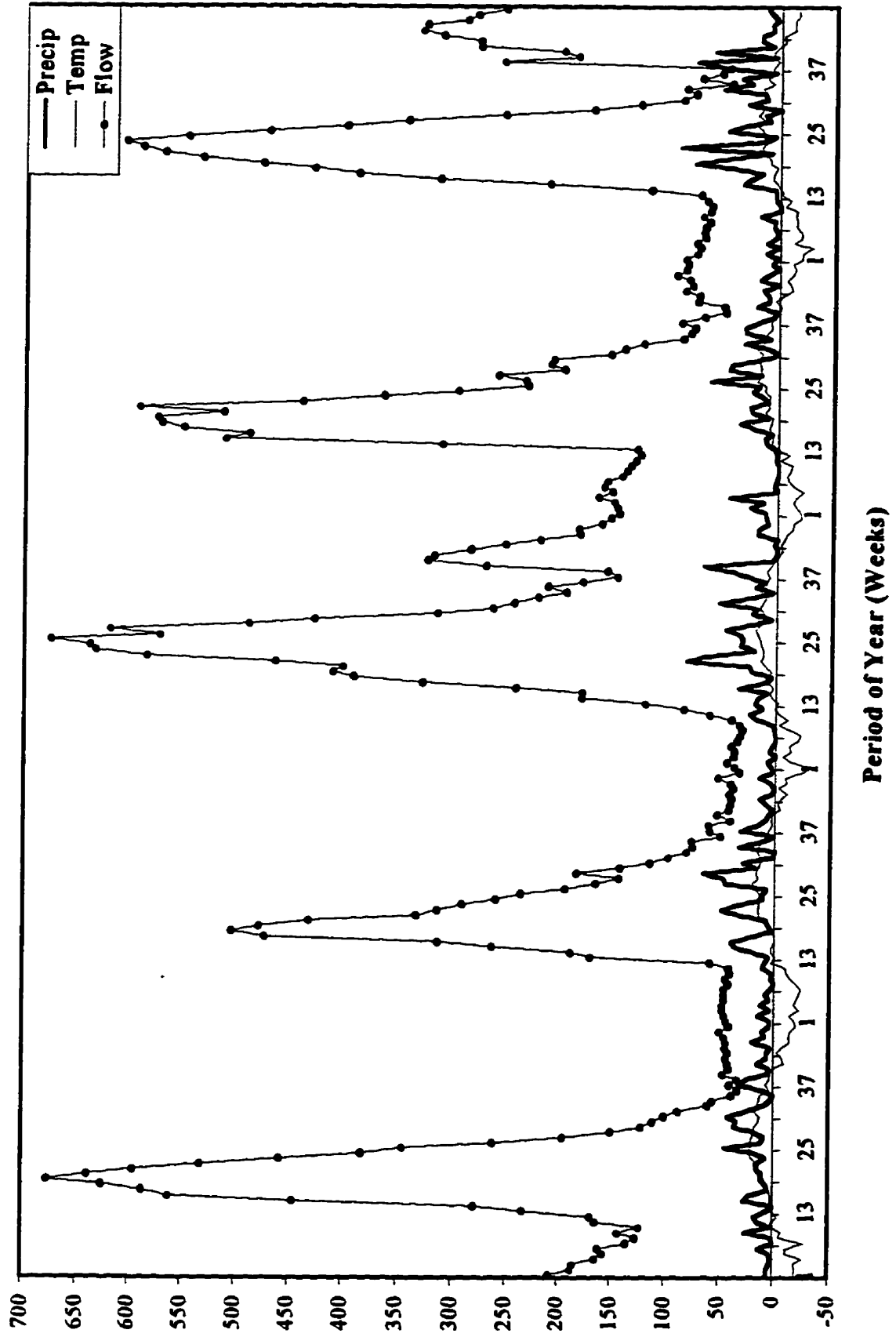
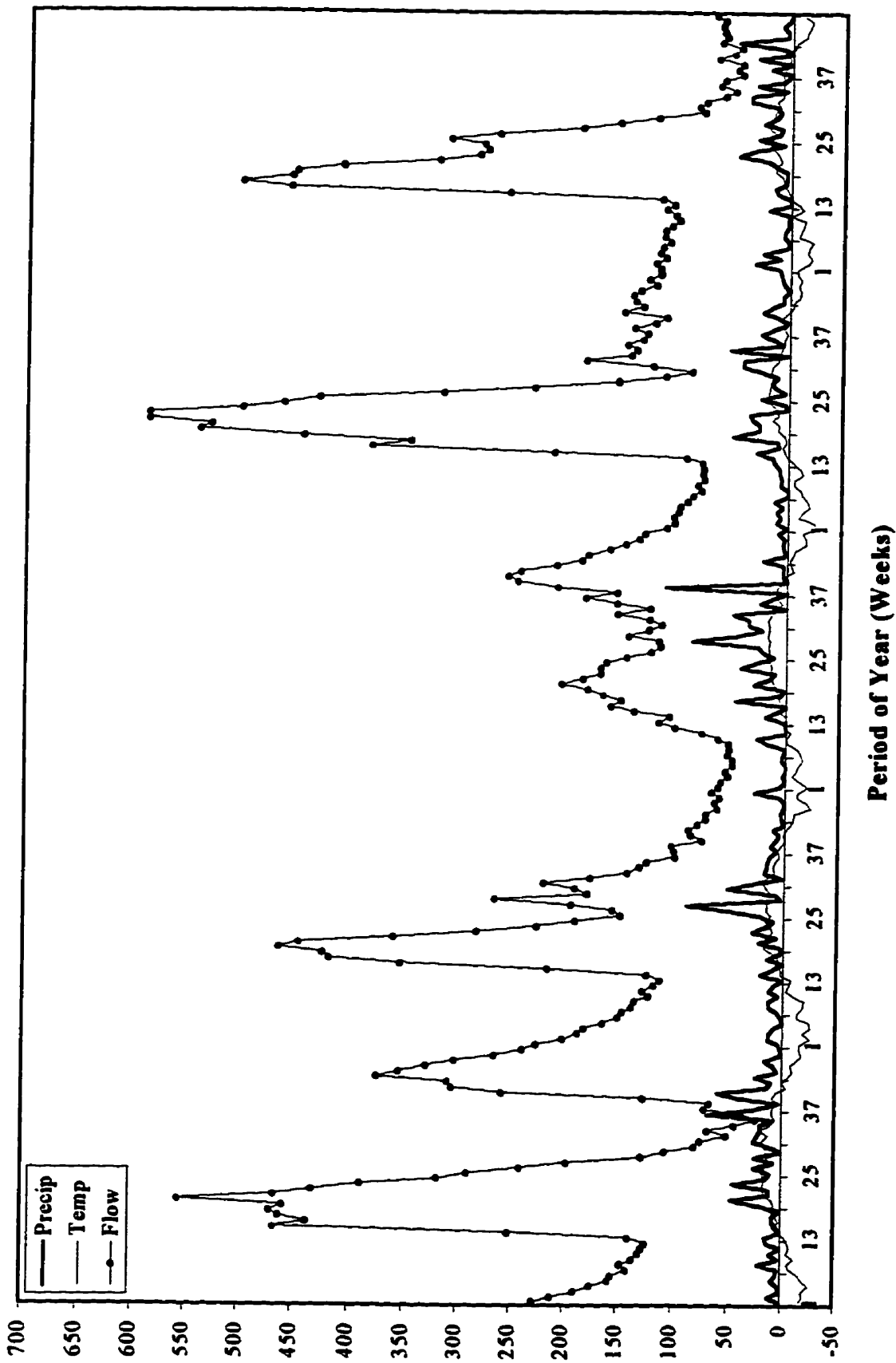


Figure A-2. Namakan Lake Historical Data, 1966-70.





Period of Year (Weeks)

Figure A-3. Namakan Lake Historical Data, 1971-75.

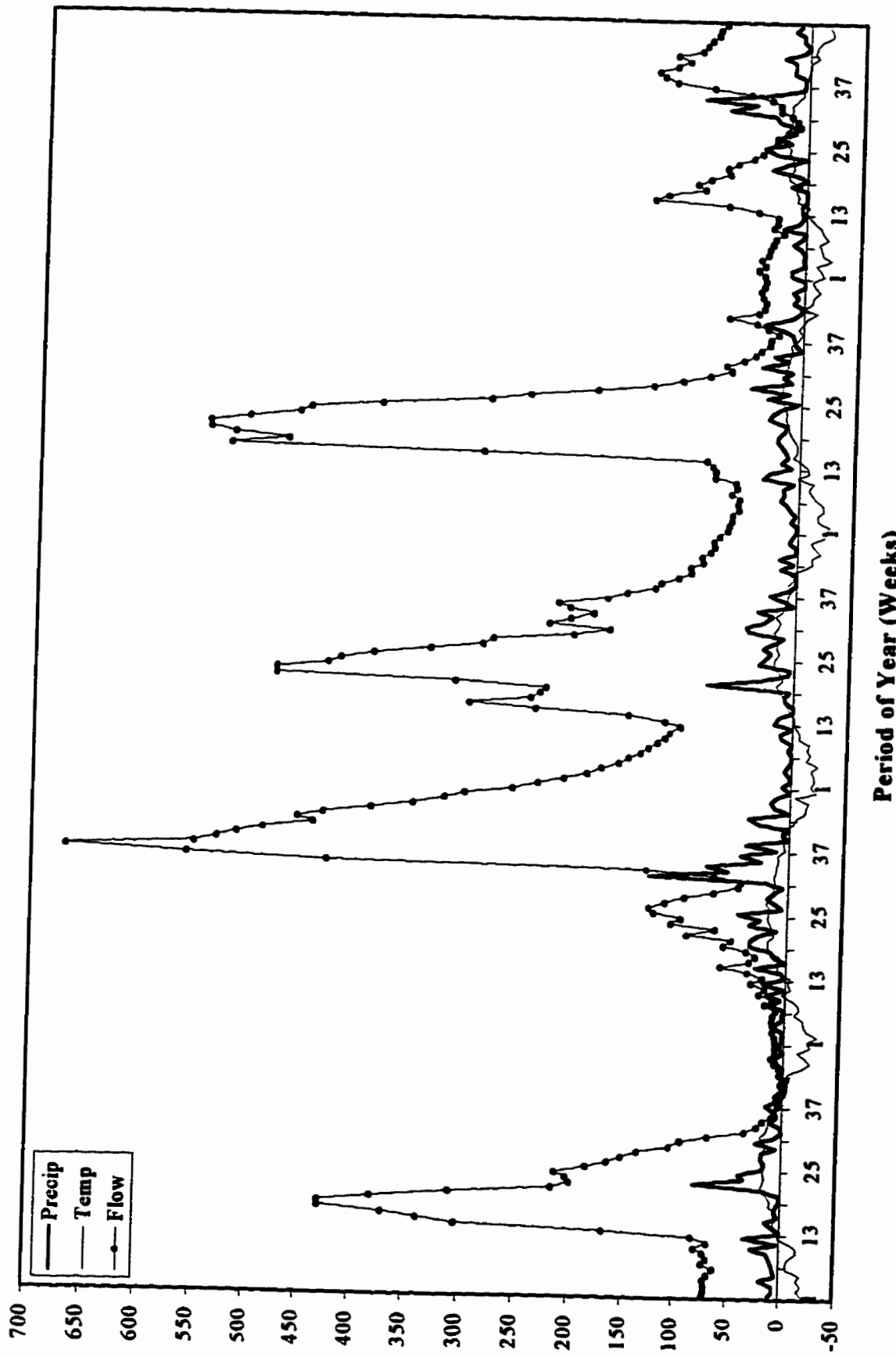
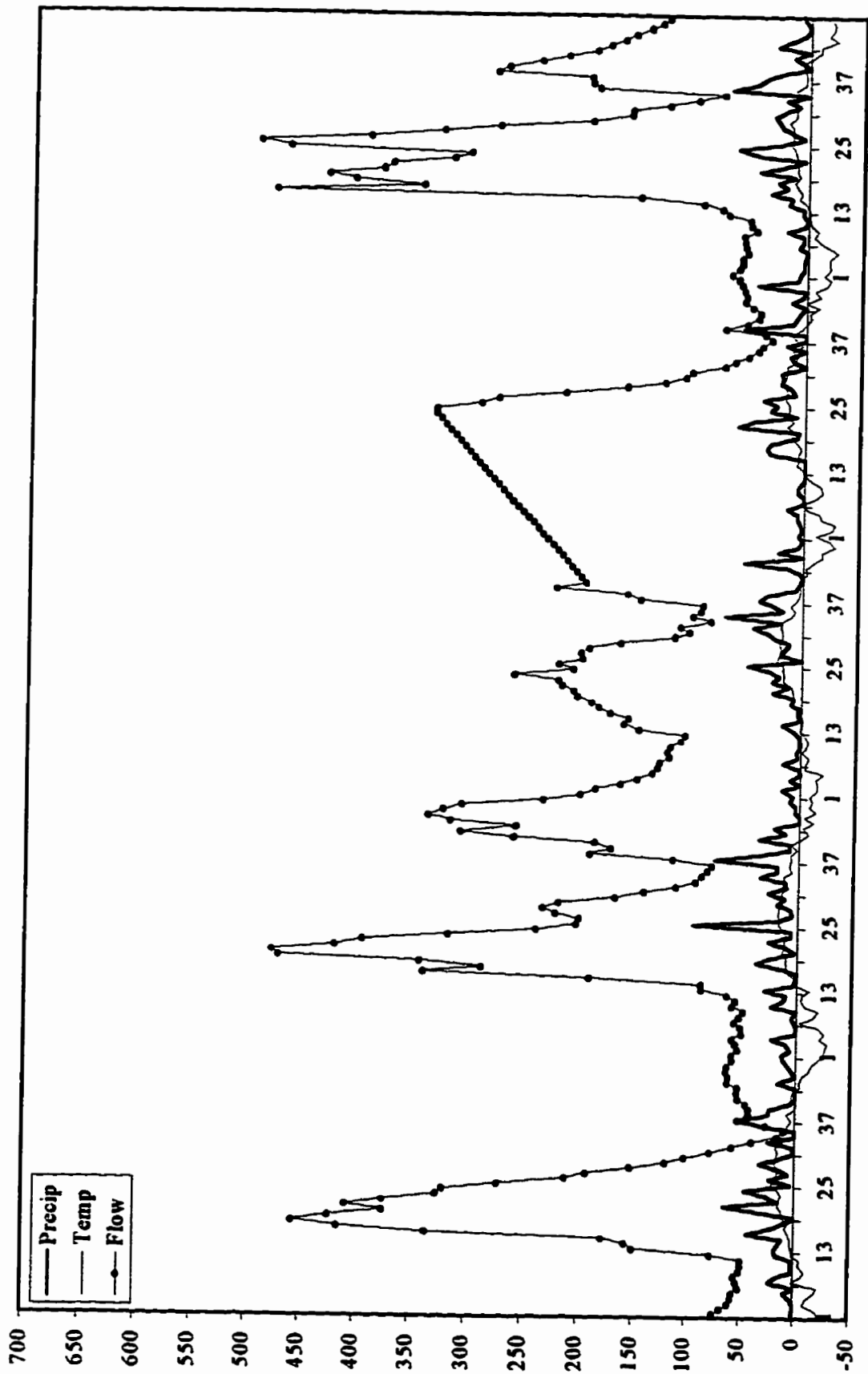
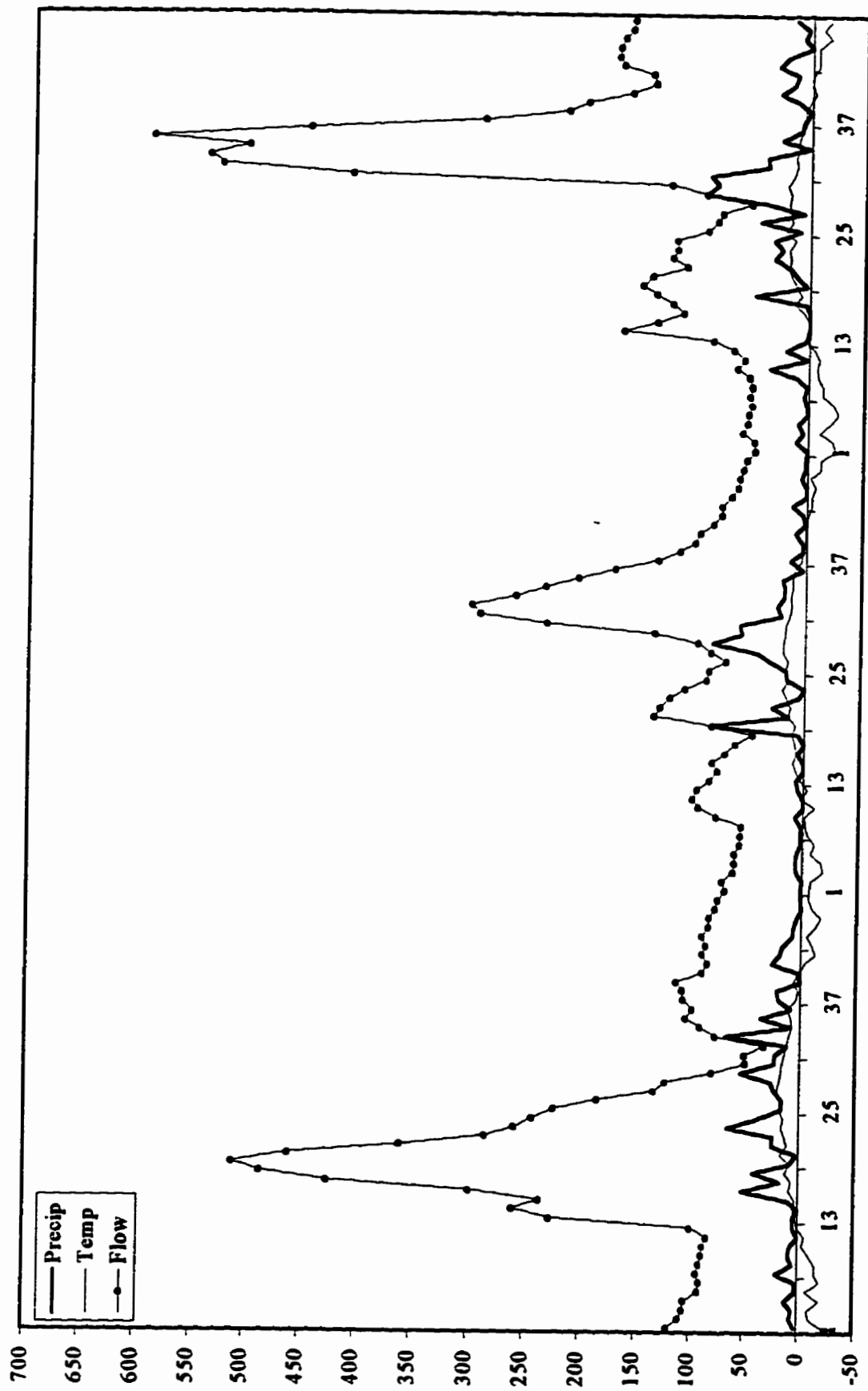


Figure A-4. Namakan Lake Historical Data, 1976-80.



Period of Year (Weeks)

Figure A-5. Namakan Lake Historical Data, 1981-85.



Period of Year (Weeks)

Figure A-6. Namakan Lake Historical Data, 1986-88.

**APPENDIX B:**  
**NAMAKAN LAKE TRAINING RESULTS, EXPERIMENT #1**

**B-1: ONE-WEEK LEAD**

**B-a-1: 1965-1969**

**B-a-2: 1970-1974**

**B-a-3: 1975-1979**

**B-a-4: 1980-1985**

**B-2: TWO-WEEK LEAD**

**B-b-1: 1965-1969**

**B-b-2: 1970-1974**

**B-b-3: 1975-1979**

**B-b-4: 1980-1985**

**B-3: THREE-WEEK LEAD**

**B-c-1: 1965-1969**

**B-c-2: 1970-1974**

**B-c-3: 1975-1979**

**B-c-4: 1980-1985**

**B-4: FOUR-WEEK LEAD**

**B-d-1: 1965-1969**

**B-d-2: 1970-1974**

**B-d-3: 1975-1979**

**B-d-4: 1980-1985**

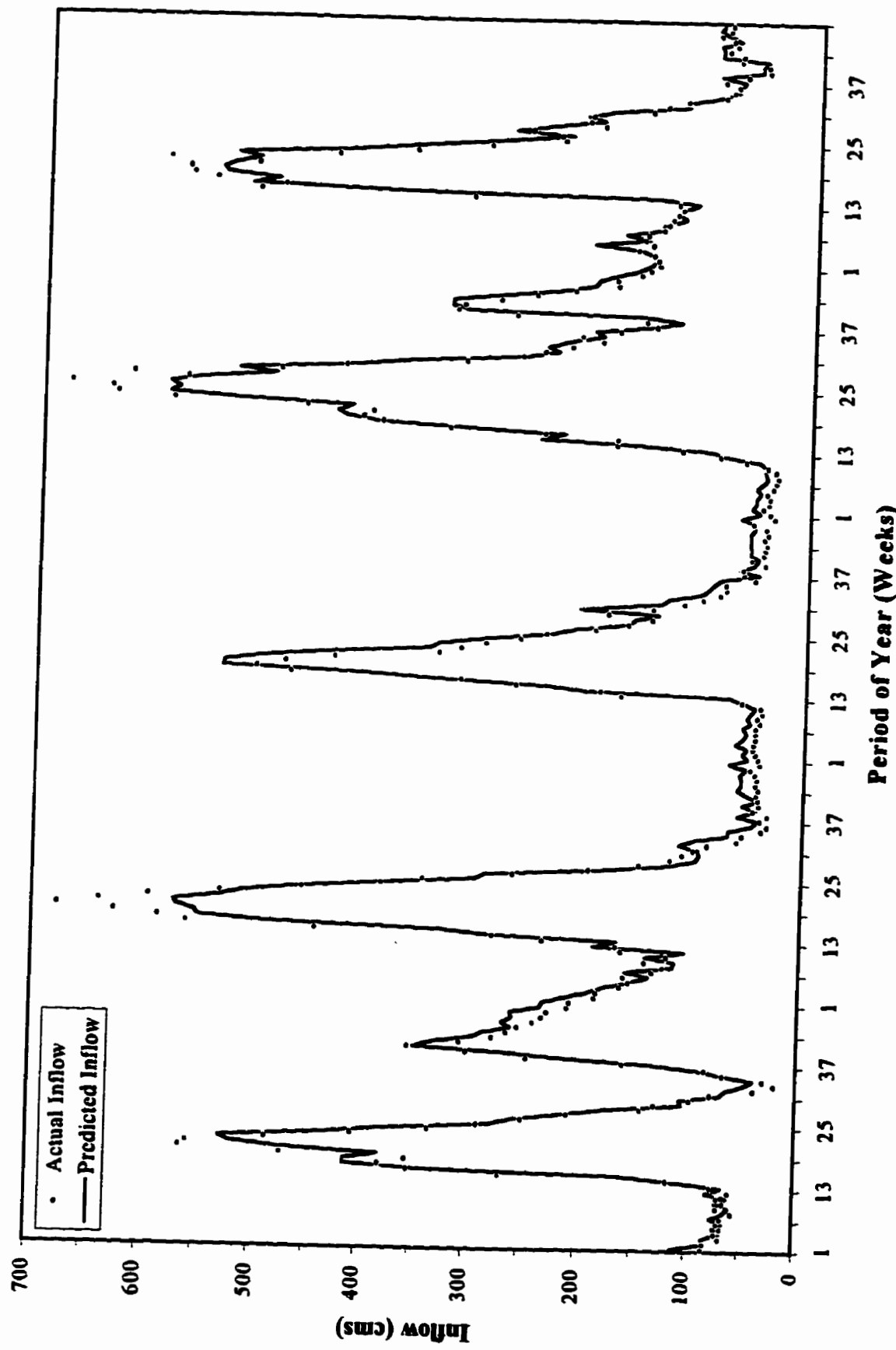


Figure B-a-1. Namakan Lake Training Results, 1965-69, one-week lead.

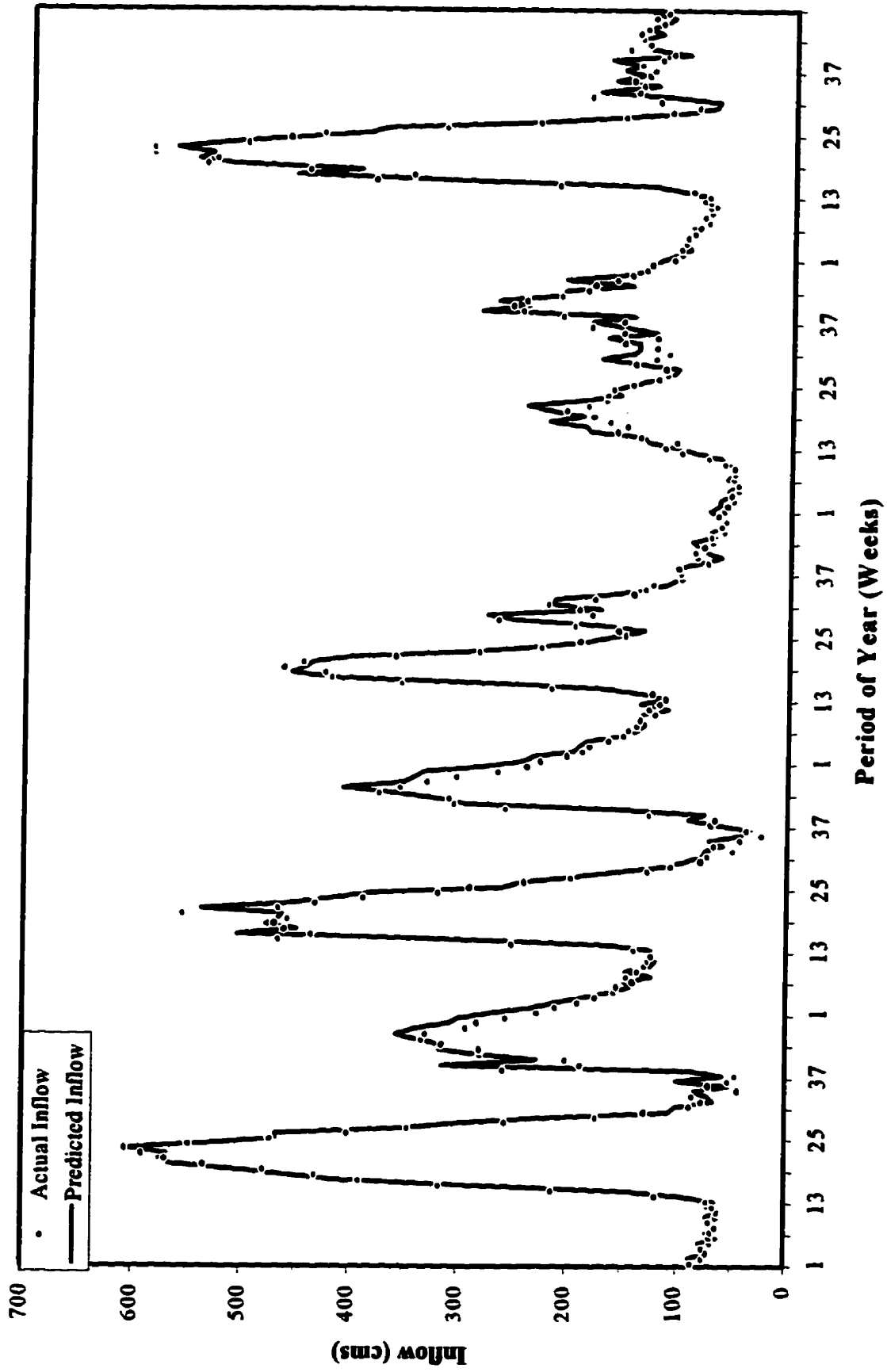


Figure B-a-2. Namakan Lake Training Results, 1970-74, one-week lead.

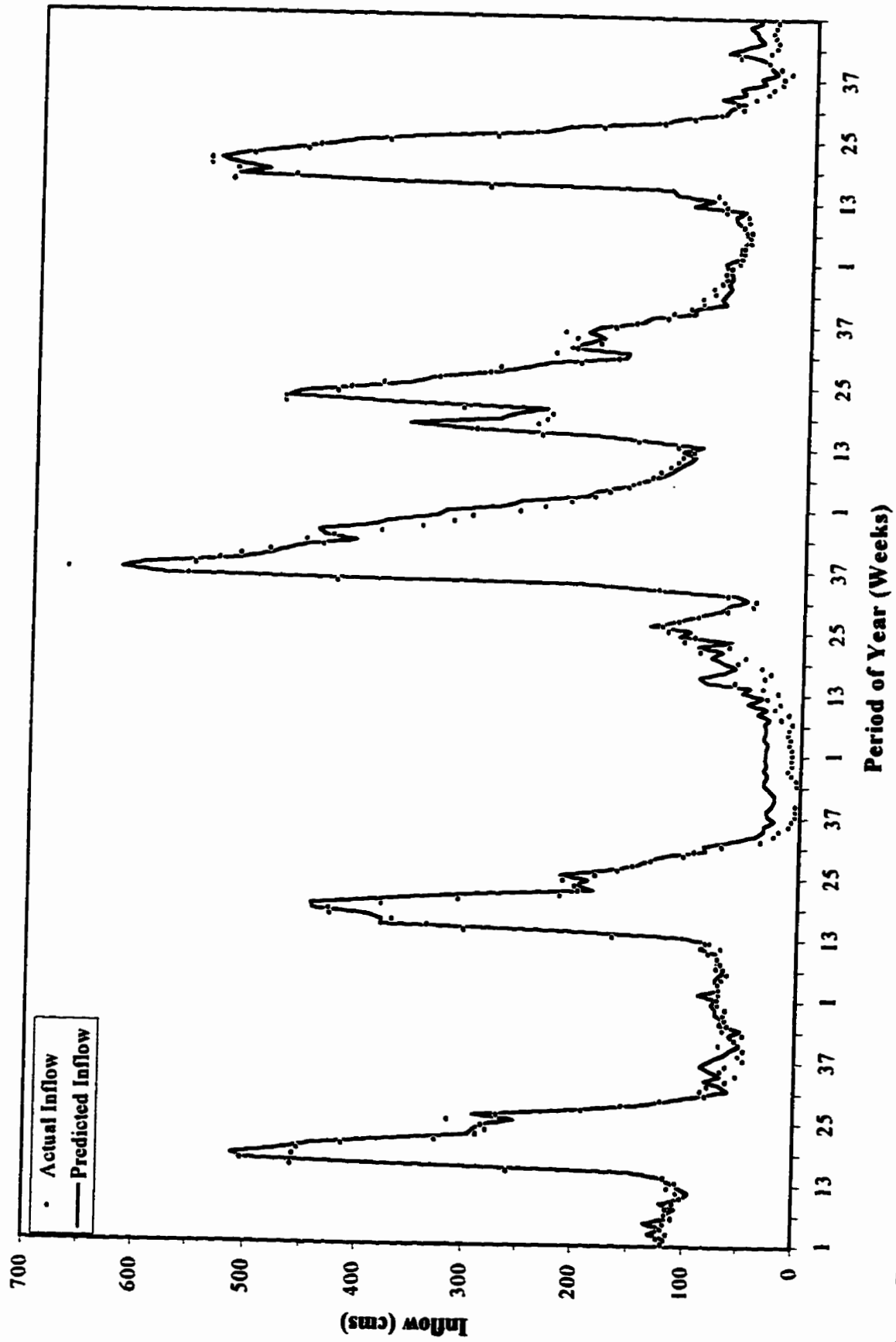


Figure B-a-3. Namakan Lake Training Results, 1975-79, one-week lead.



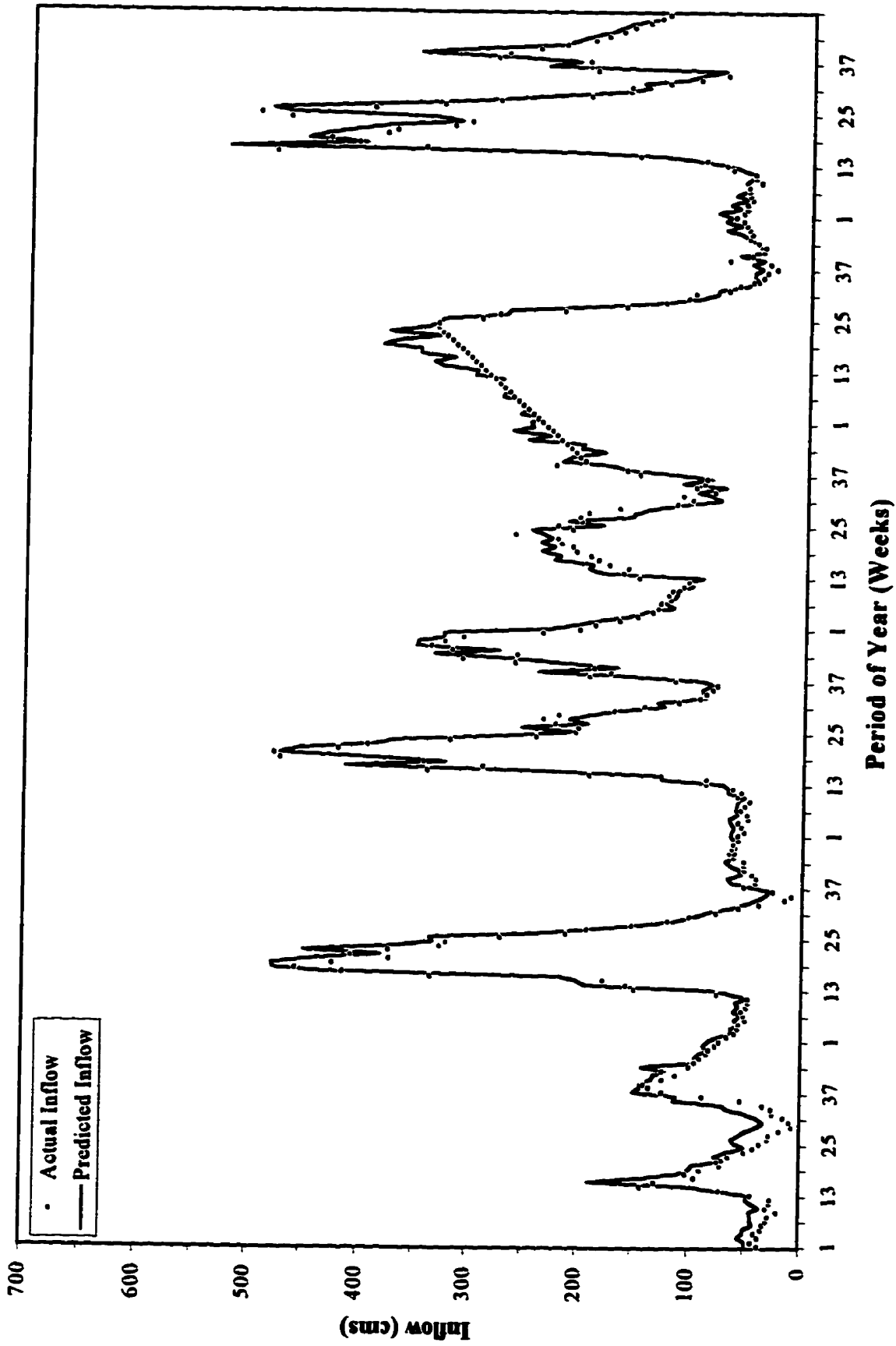


Figure B-a-4. Namakan Lake Training Results, 1980-85, one-week lead.

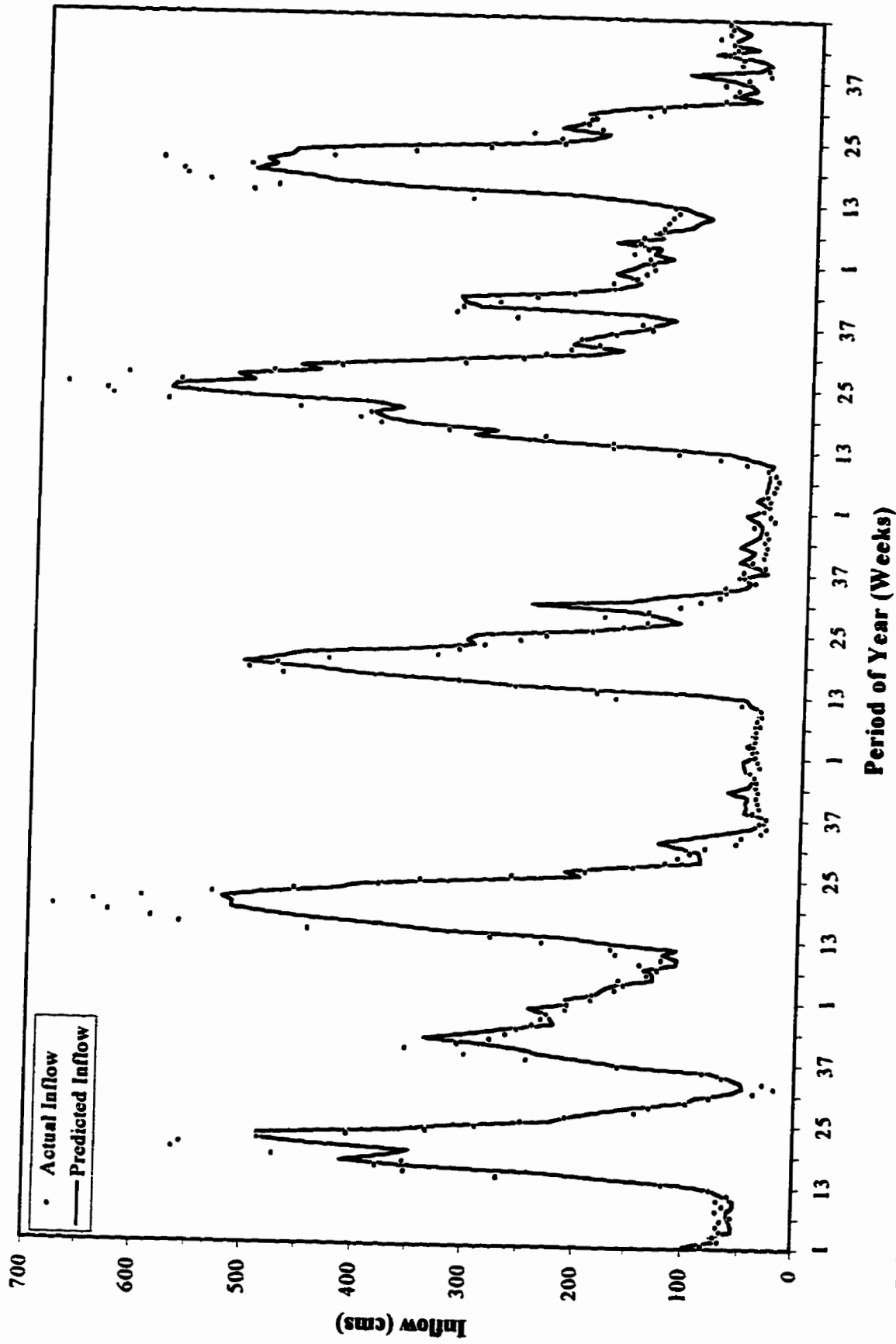
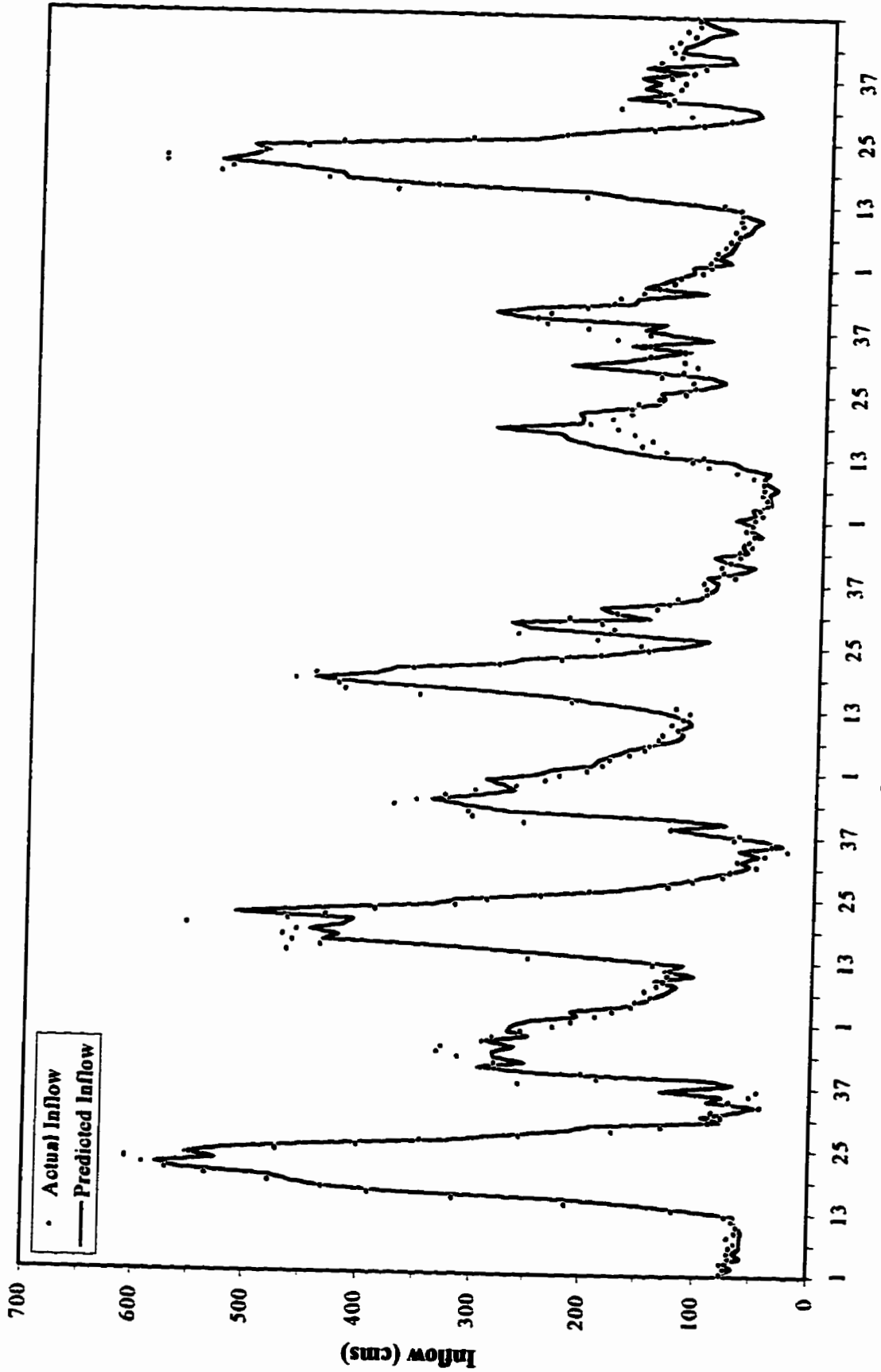


Figure B-b-1. Namakan Lake Training Results, 1965-69, two-week lead.



**Figure B-b-2. Namakan Lake Training Results, 1970-74, two-week lead.**

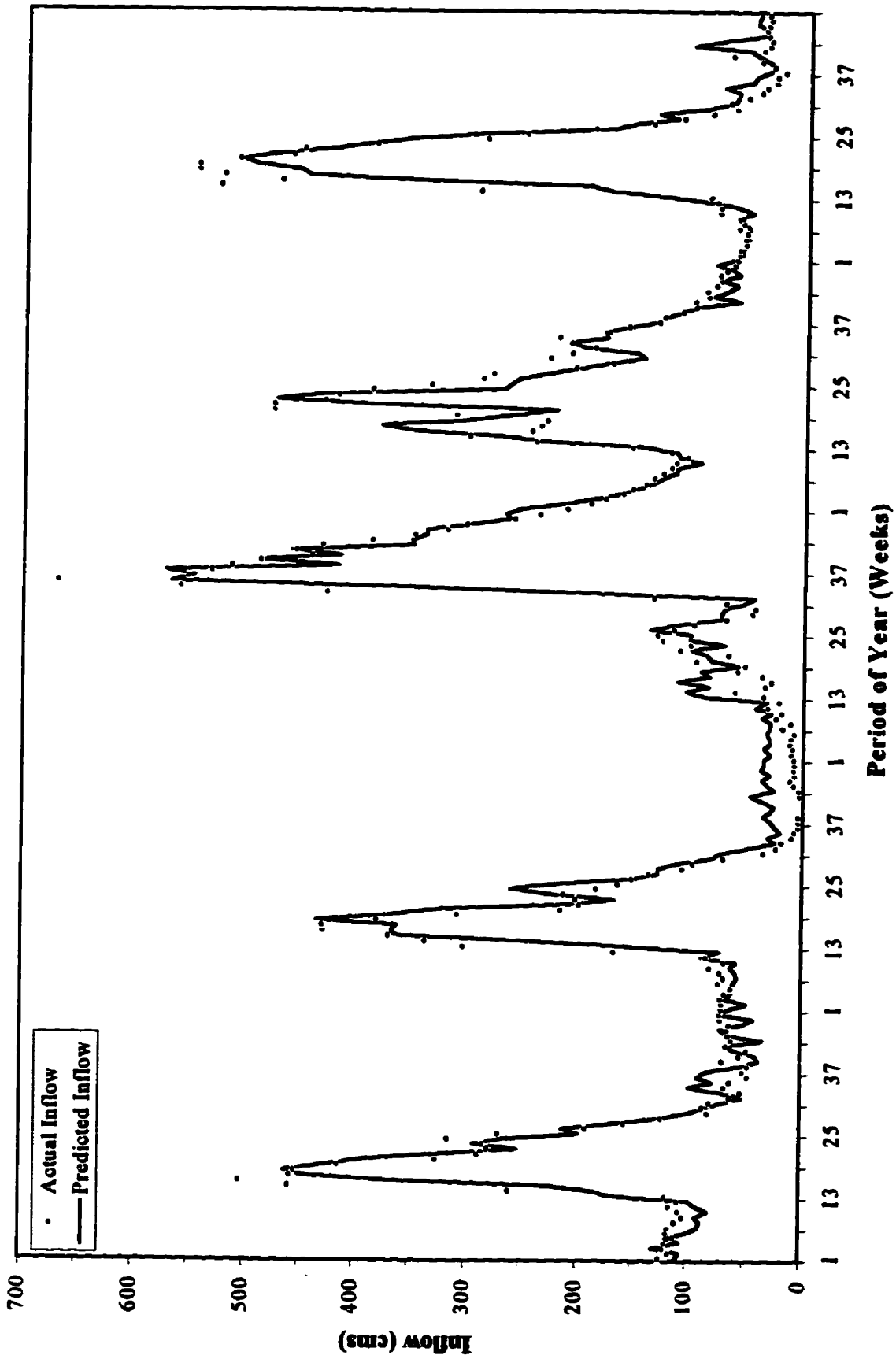


Figure B-b-3. Namakan Lake Training Results, 1975-79, two-week lead.

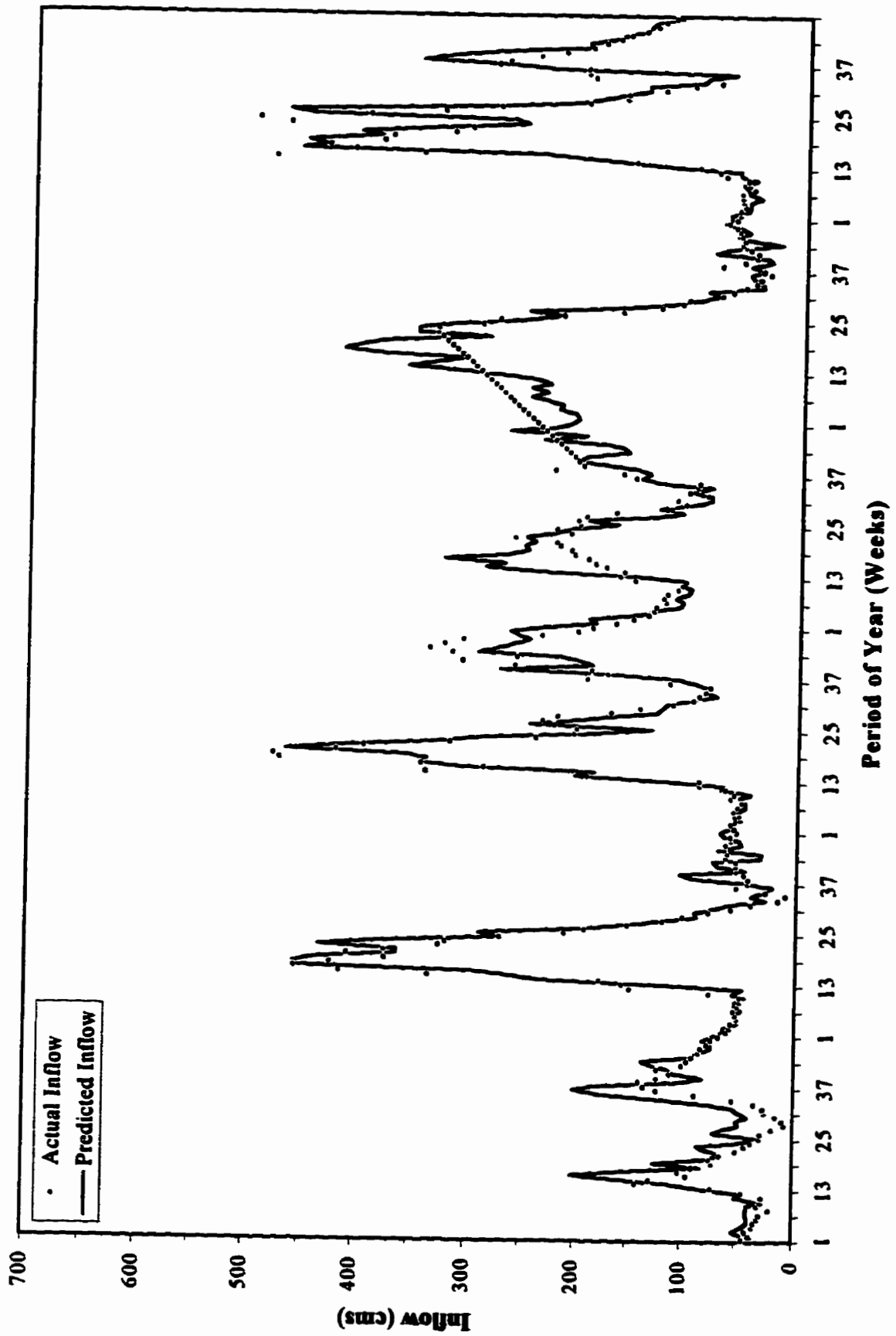


Figure B-b-4. Namakan Lake Training Results, 1980-85, two-week lead.

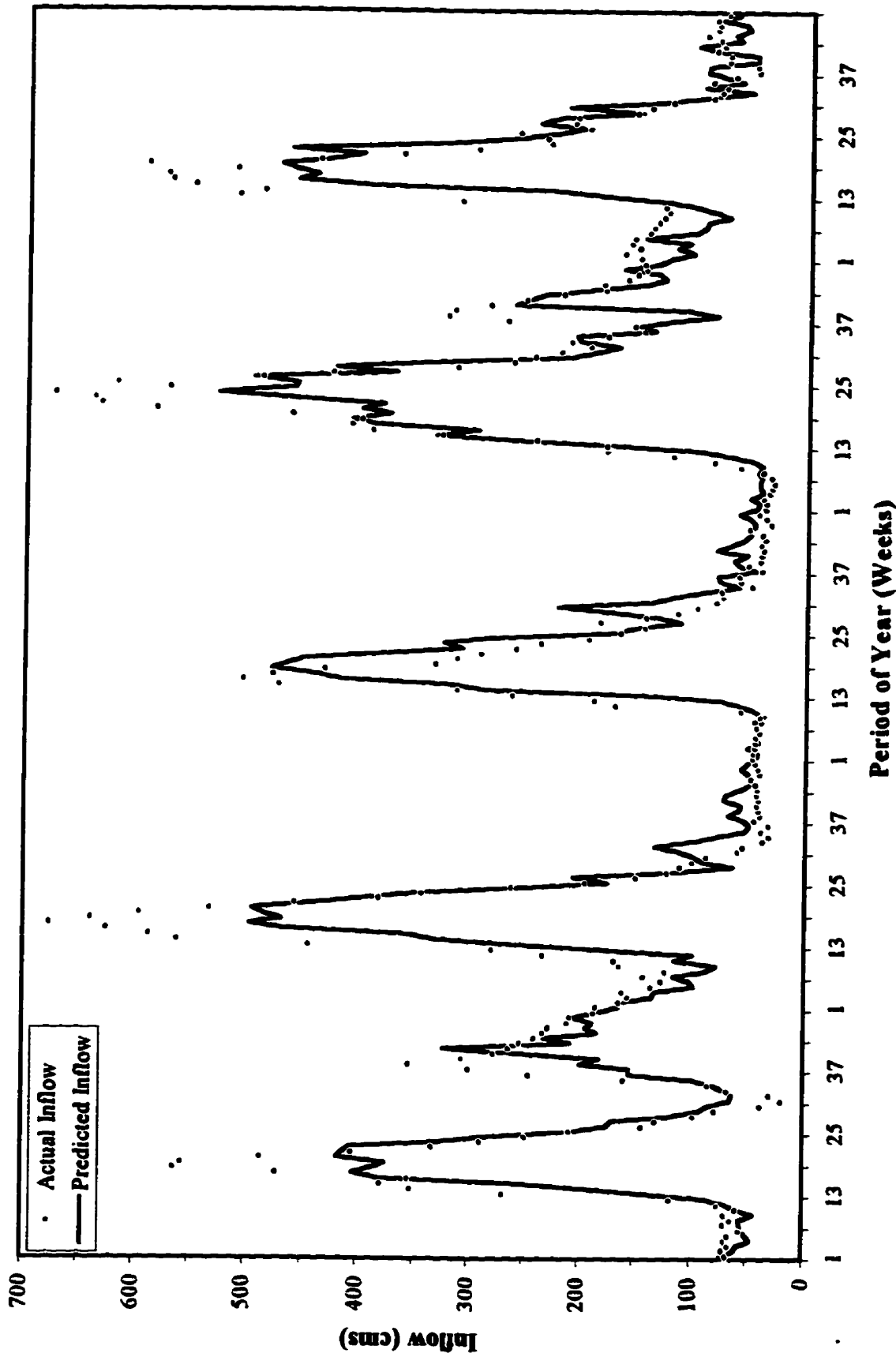


Figure B-c-1. Namakan Lake Training Results, 1965-69, three-week lead.

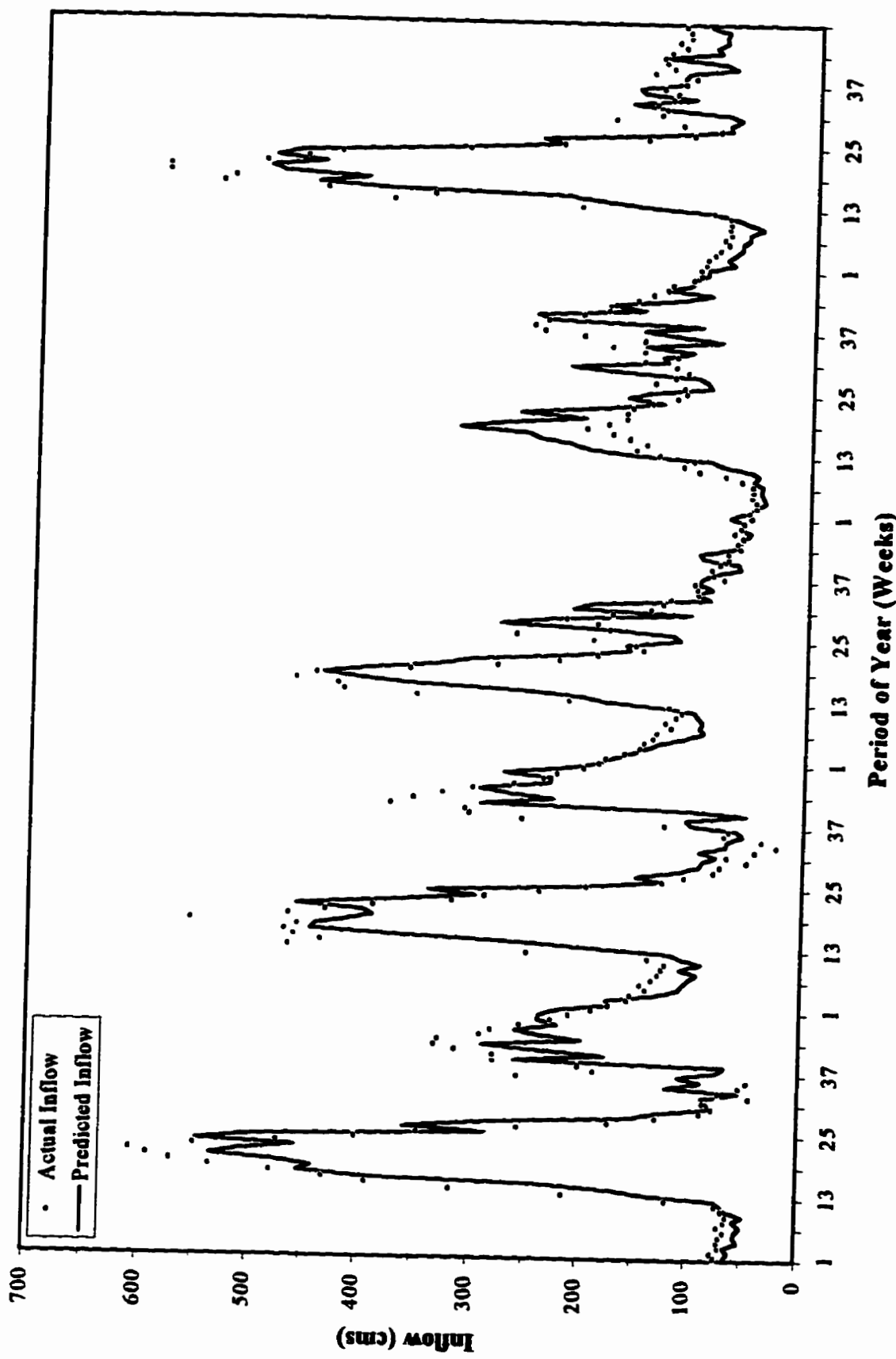


Figure B-c-2. Namakan Lake Training Results, 1970-74, three-week lead.

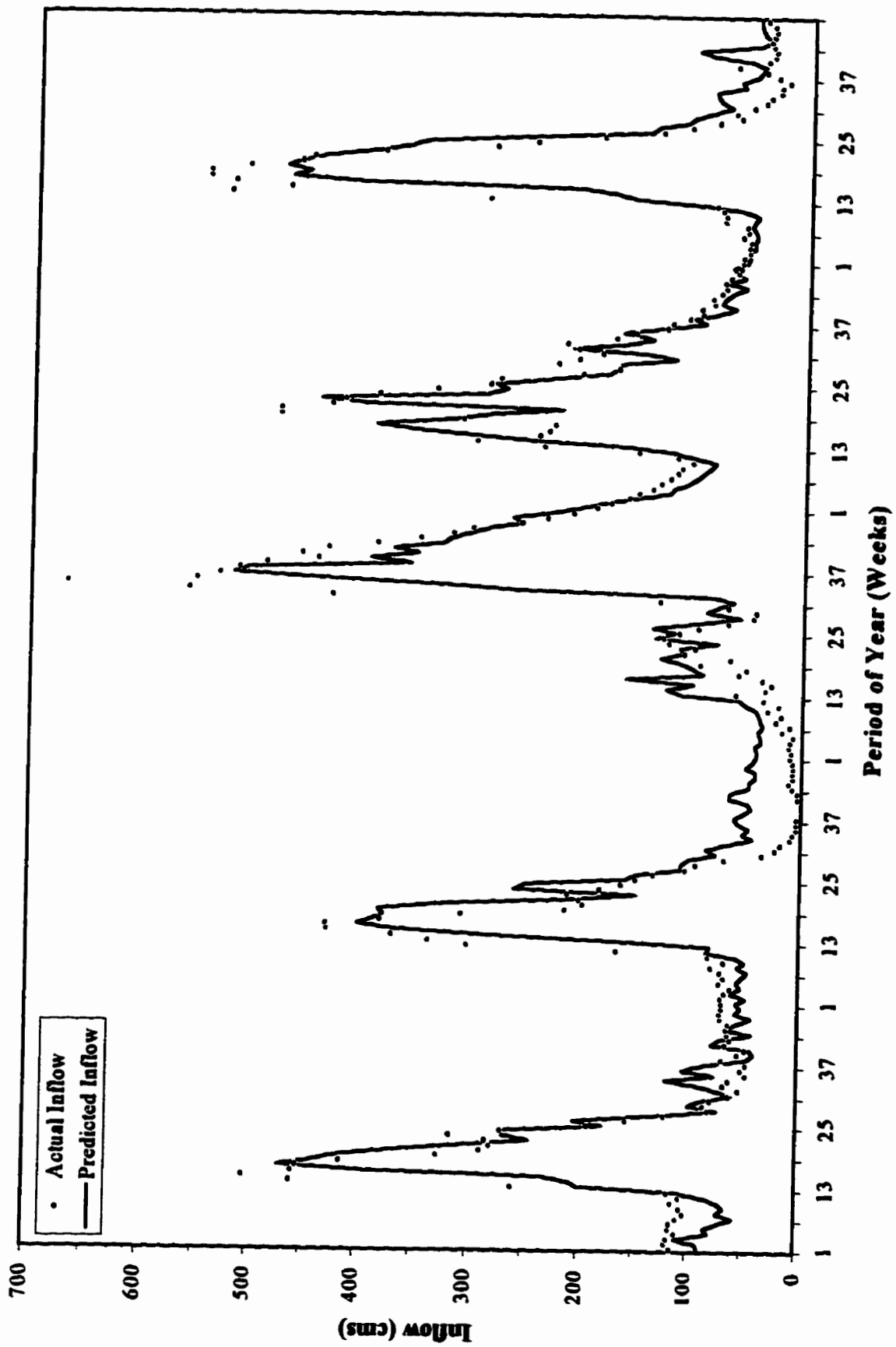


Figure B-c-3. Namakan Lake Training Results, 1975-79, three-week lead.



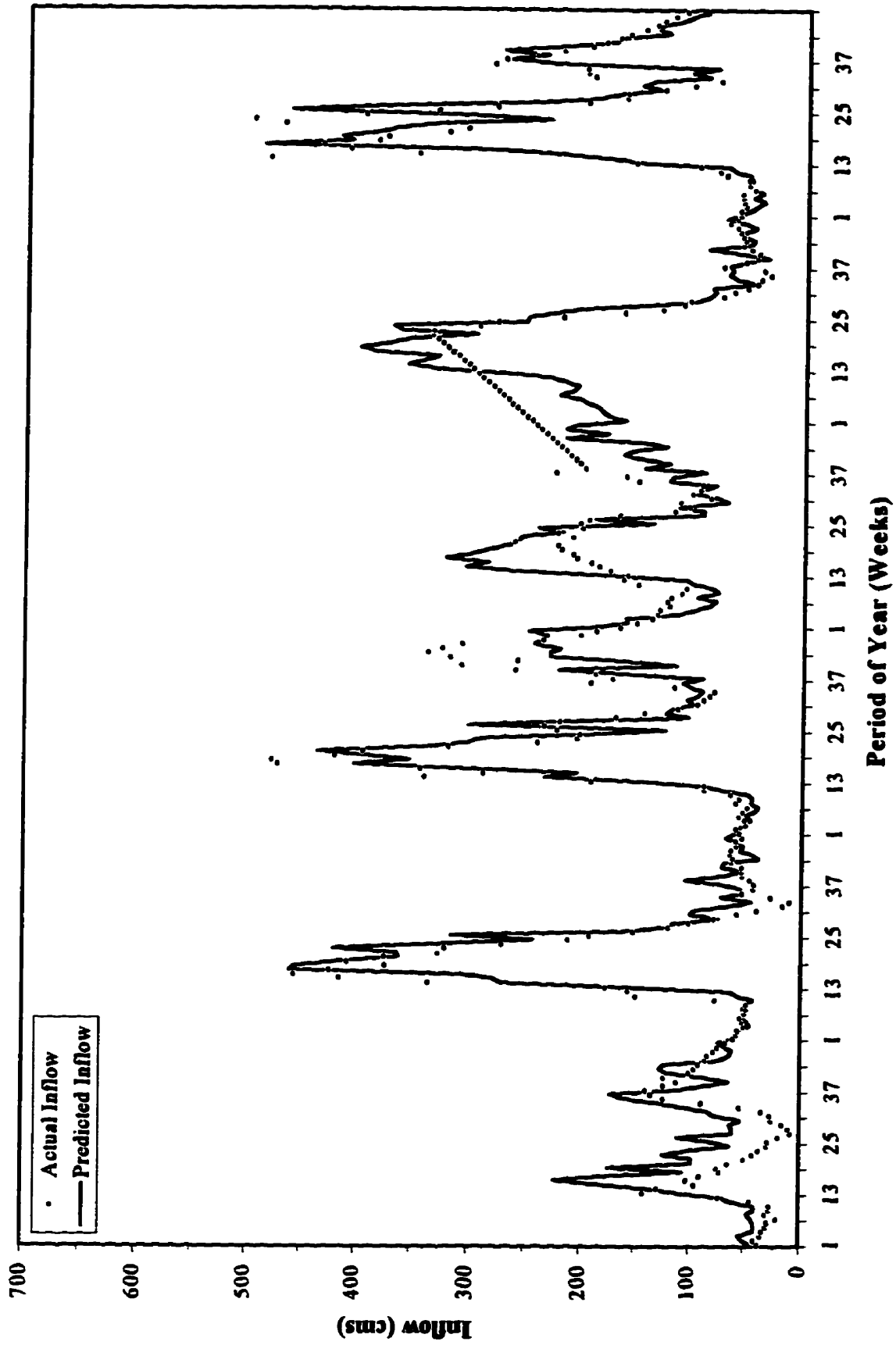


Figure B-c-4. Namakan Lake Training Results, 1980-85, three-week lead.

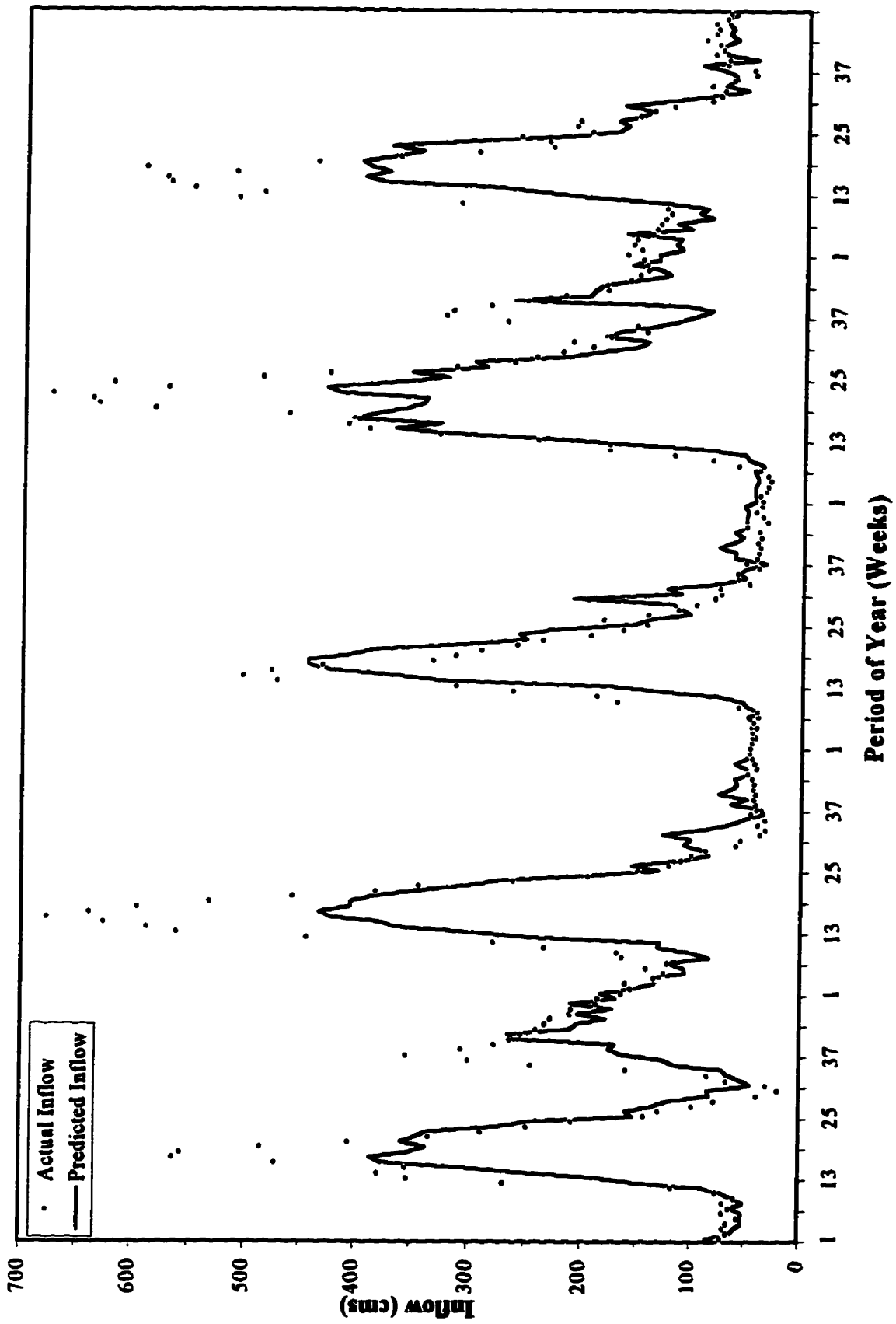


Figure B-d-1. Namakan Lake Training Results, 1965-69, four-week lead.

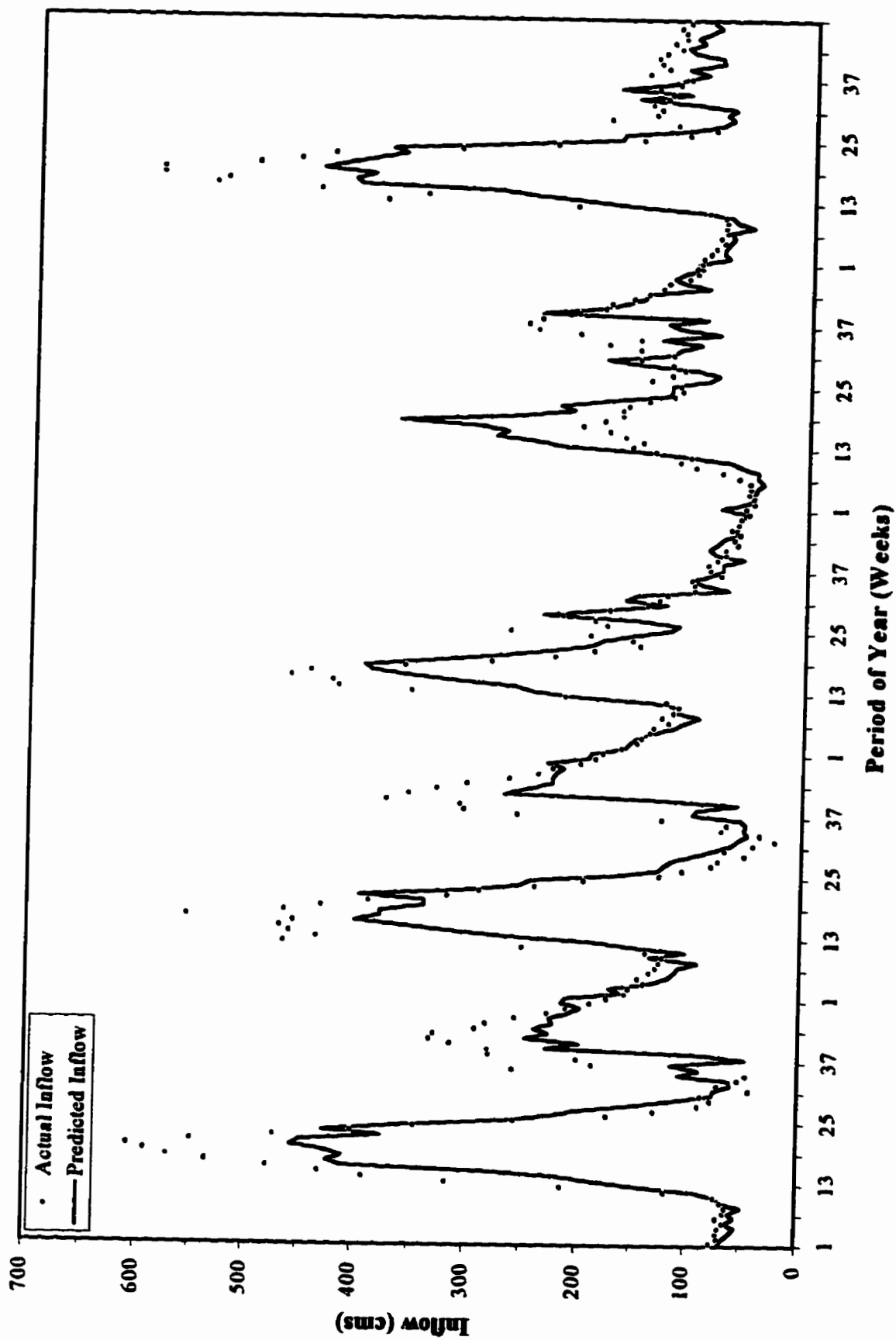


Figure B-d-2. Namakan Lake Training Results, 1970-74, four-week lead.

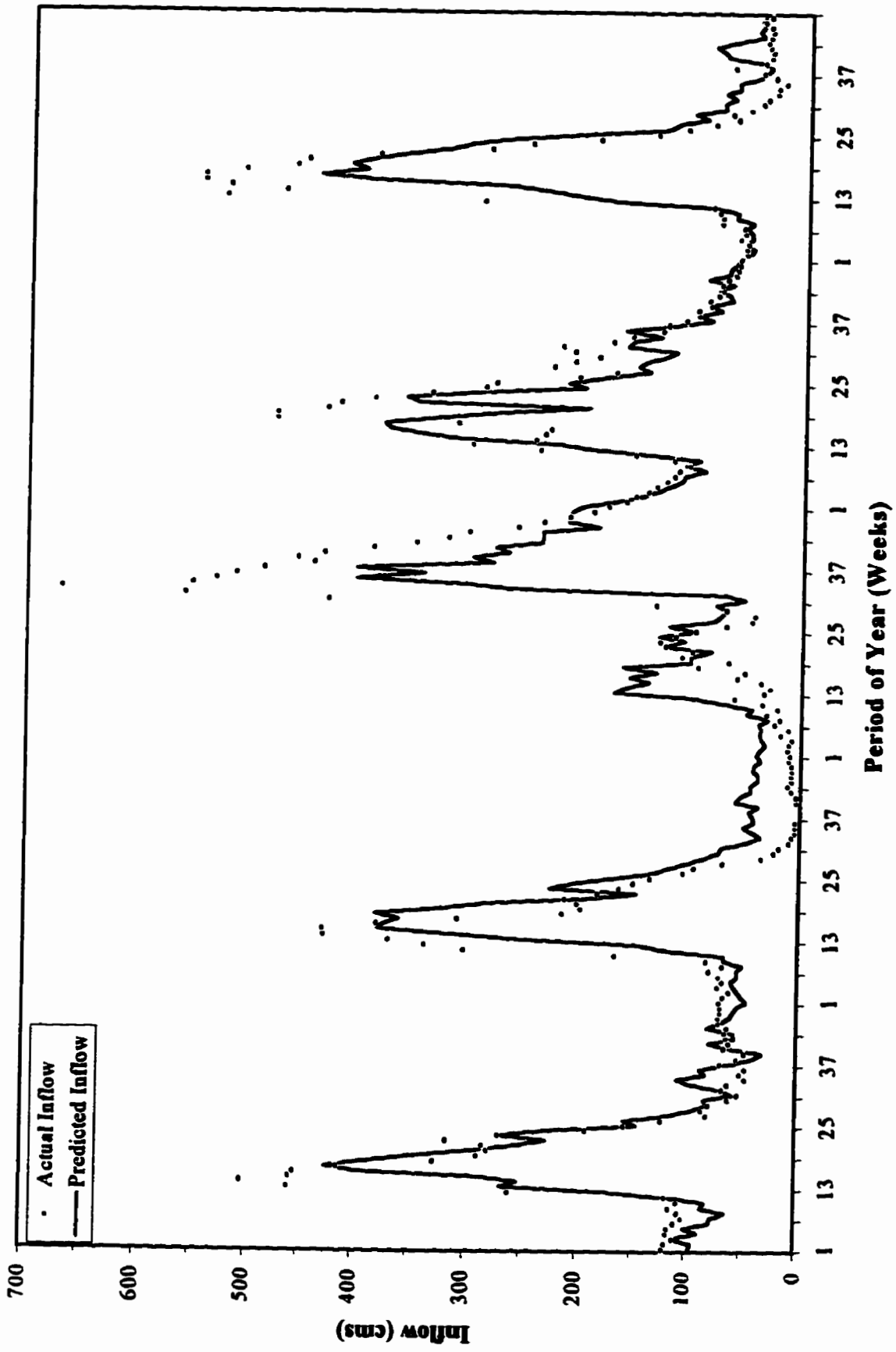


Figure B-d-3. Namakan Lake Training Results, 1975-79, four-week lead.

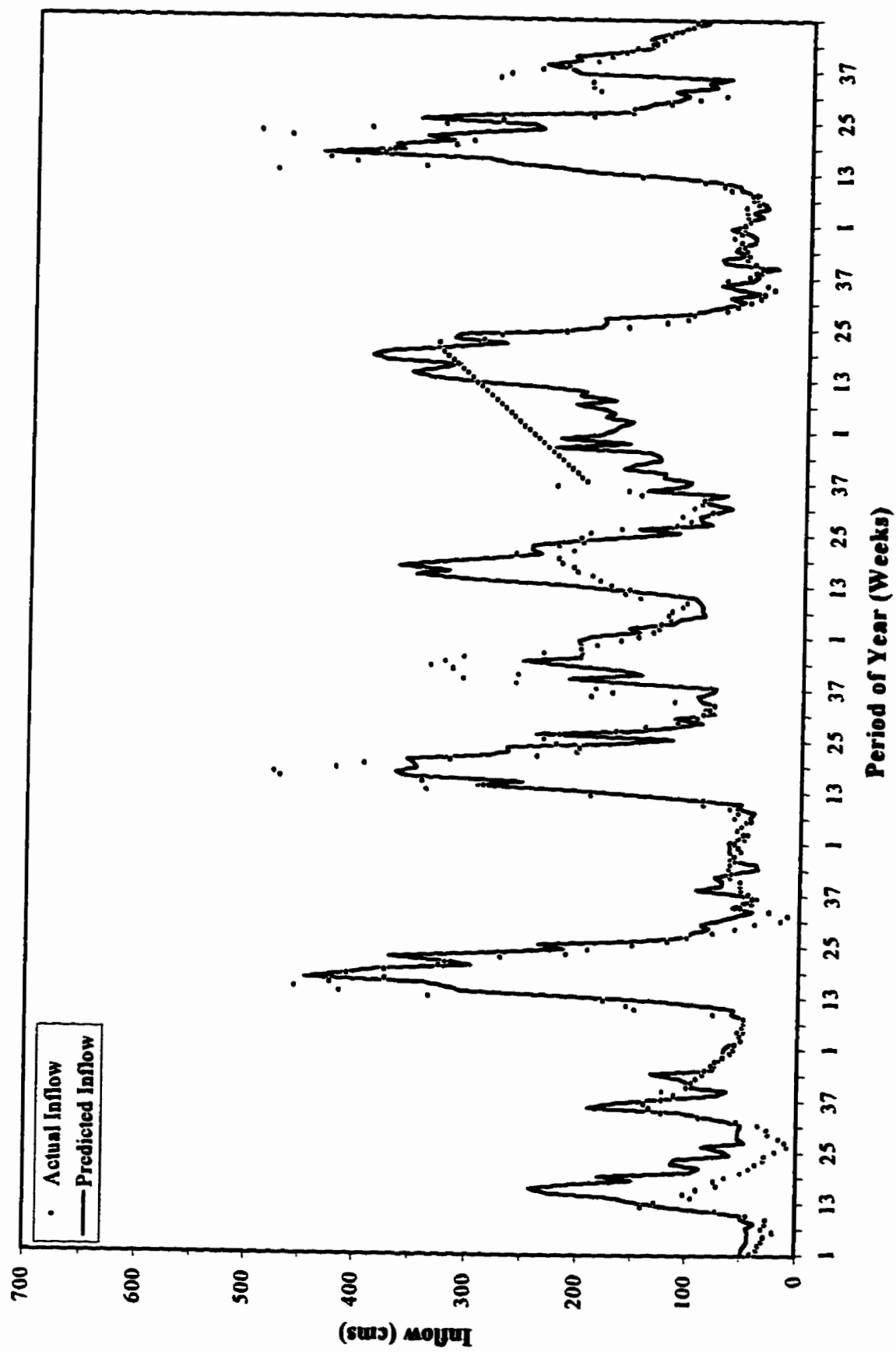


Figure B-d-4. Namakan Lake Training Results, 1980-85, four-week lead.

**APPENDIX C:**

**NAMAKAN LAKE TRAINING RESULTS, EXPERIMENT #2**

**C-1: ONE-WEEK LEAD**

**C-a-1: 1965-1969**

**C-a-2: 1970-1974**

**C-a-3: 1975-1979**

**C-a-4: 1980-1985**

**C-2: TWO-WEEK LEAD**

**C-b-1: 1965-1969**

**C-b-2: 1970-1974**

**C-b-3: 1975-1979**

**C-b-4: 1980-1985**

**C-3: THREE-WEEK LEAD**

**C-c-1: 1965-1969**

**C-c-2: 1970-1974**

**C-c-3: 1975-1979**

**C-c-4: 1980-1985**

**C-4: FOUR-WEEK LEAD**

**C-d-1: 1965-1969**

**C-d-2: 1970-1974**

**C-d-3: 1975-1979**

**C-d-4: 1980-1985**

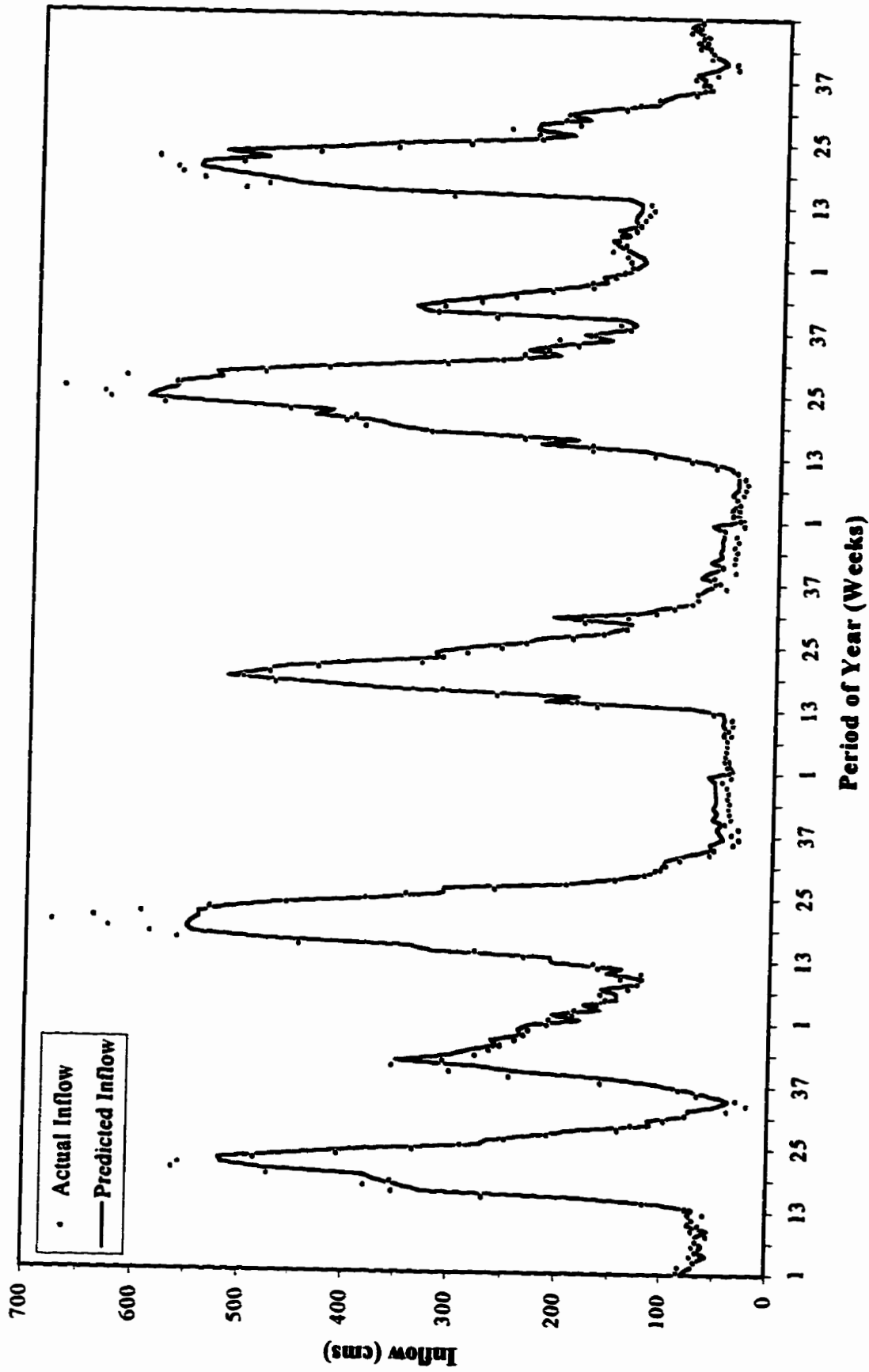


Figure C-a-1. Namakan Lake Training Results, 1965-69, one-week lead.

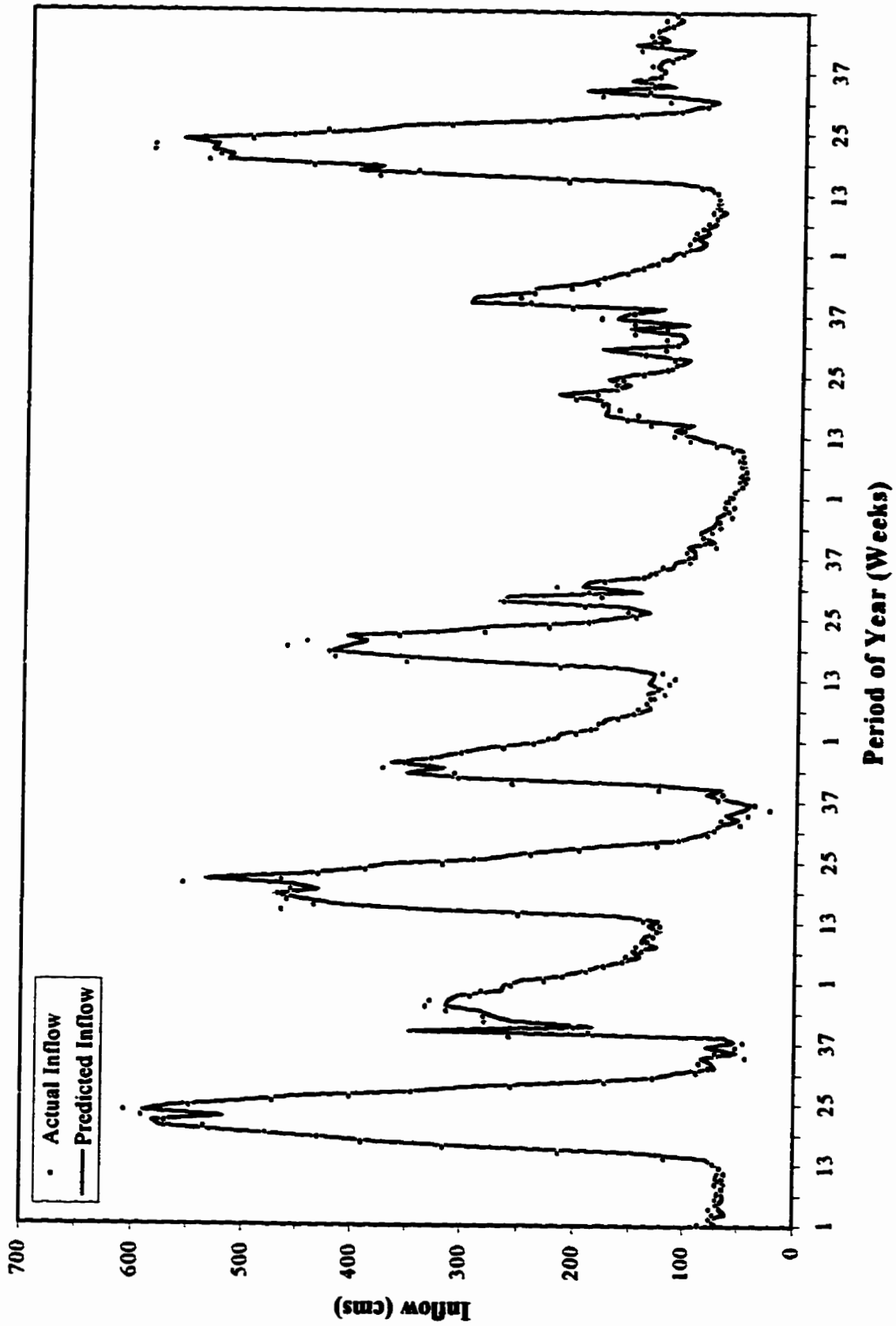


Figure C-a-2. Namakan Lake Training Results, 1970-74, one-week lead.



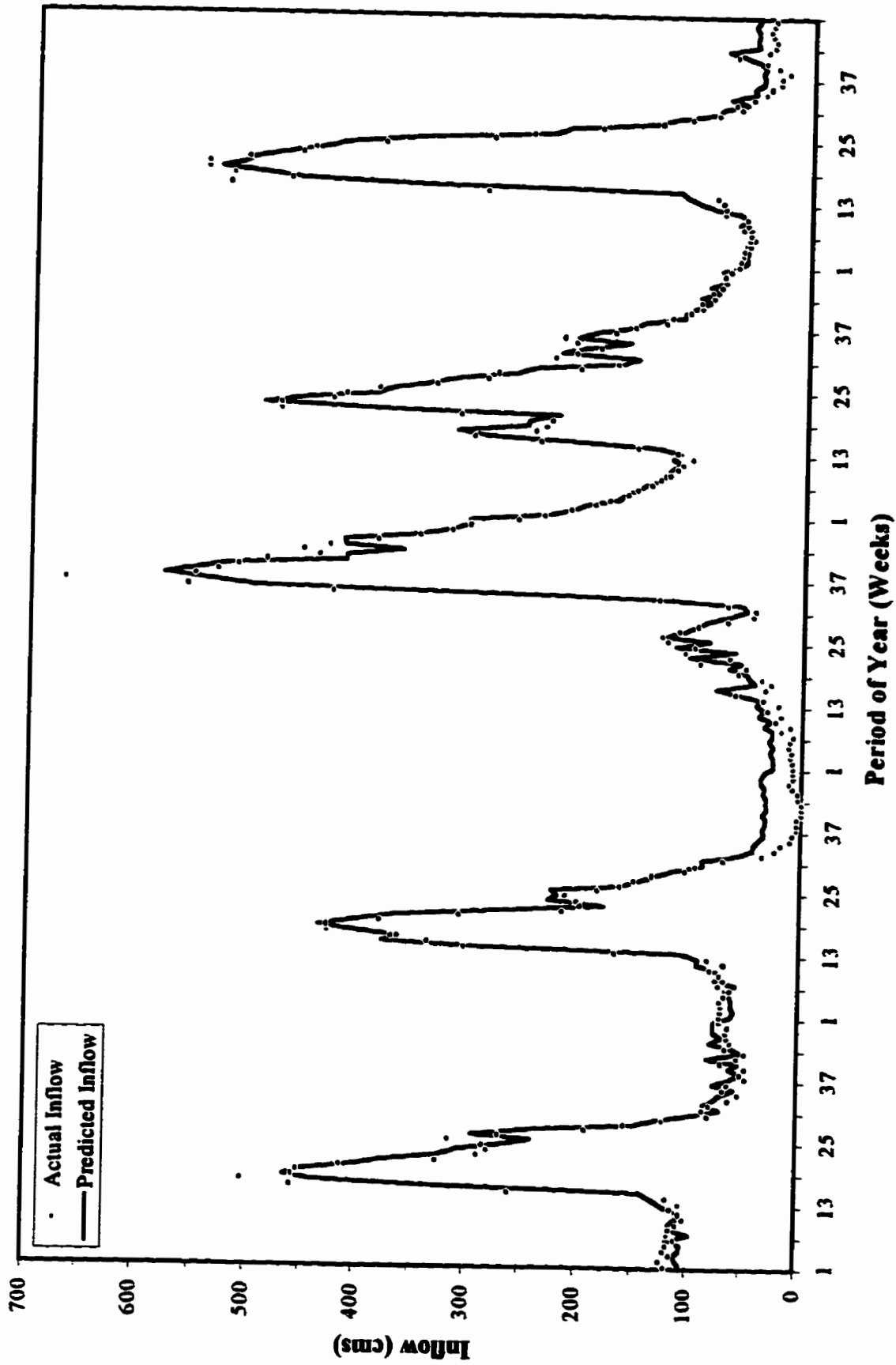


Figure C-a-3. Namakan Lake Training Results, 1975-79, one-week lead.

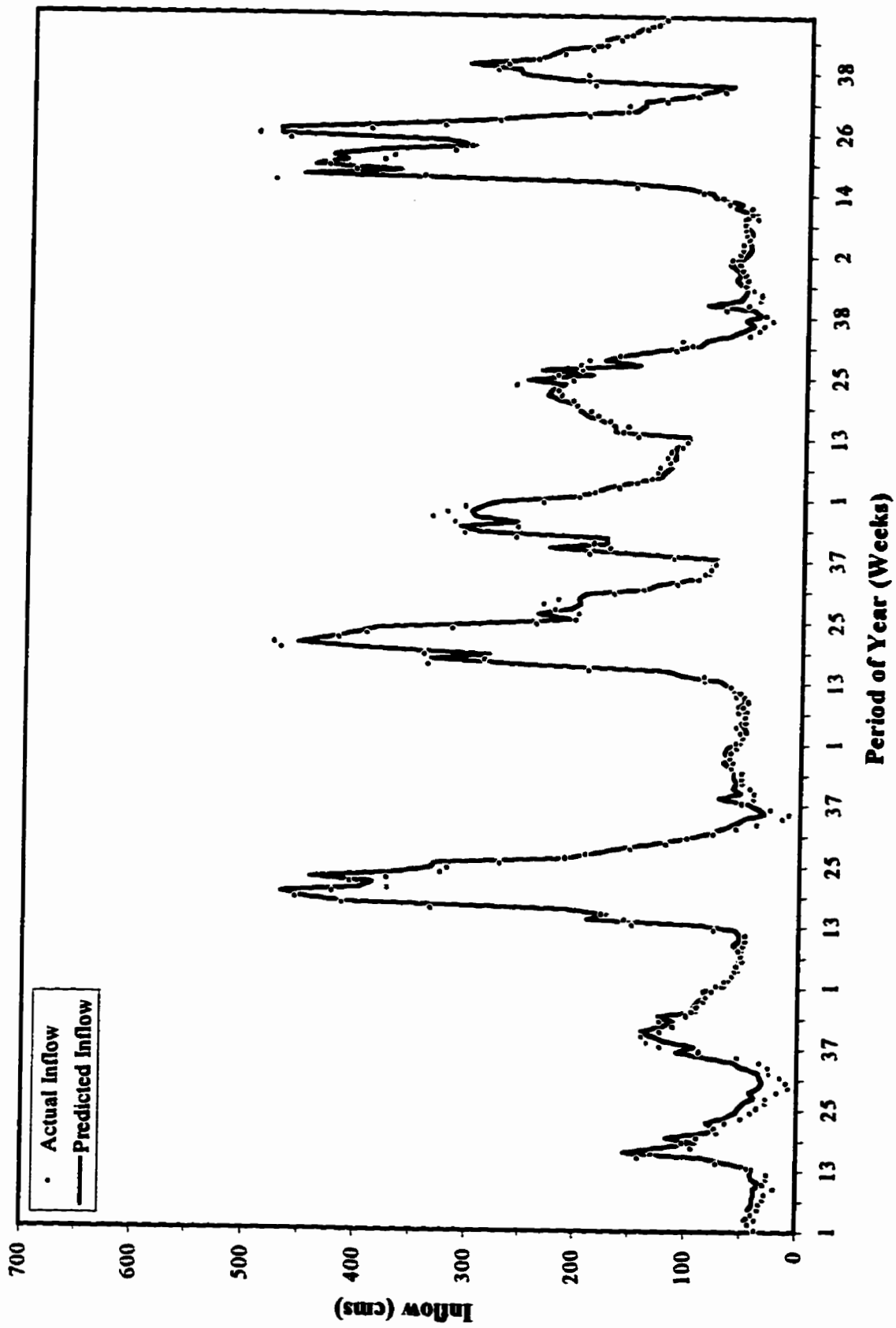


Figure C-a-4. Namakan Lake Training Results, 1980-85, one-week lead.

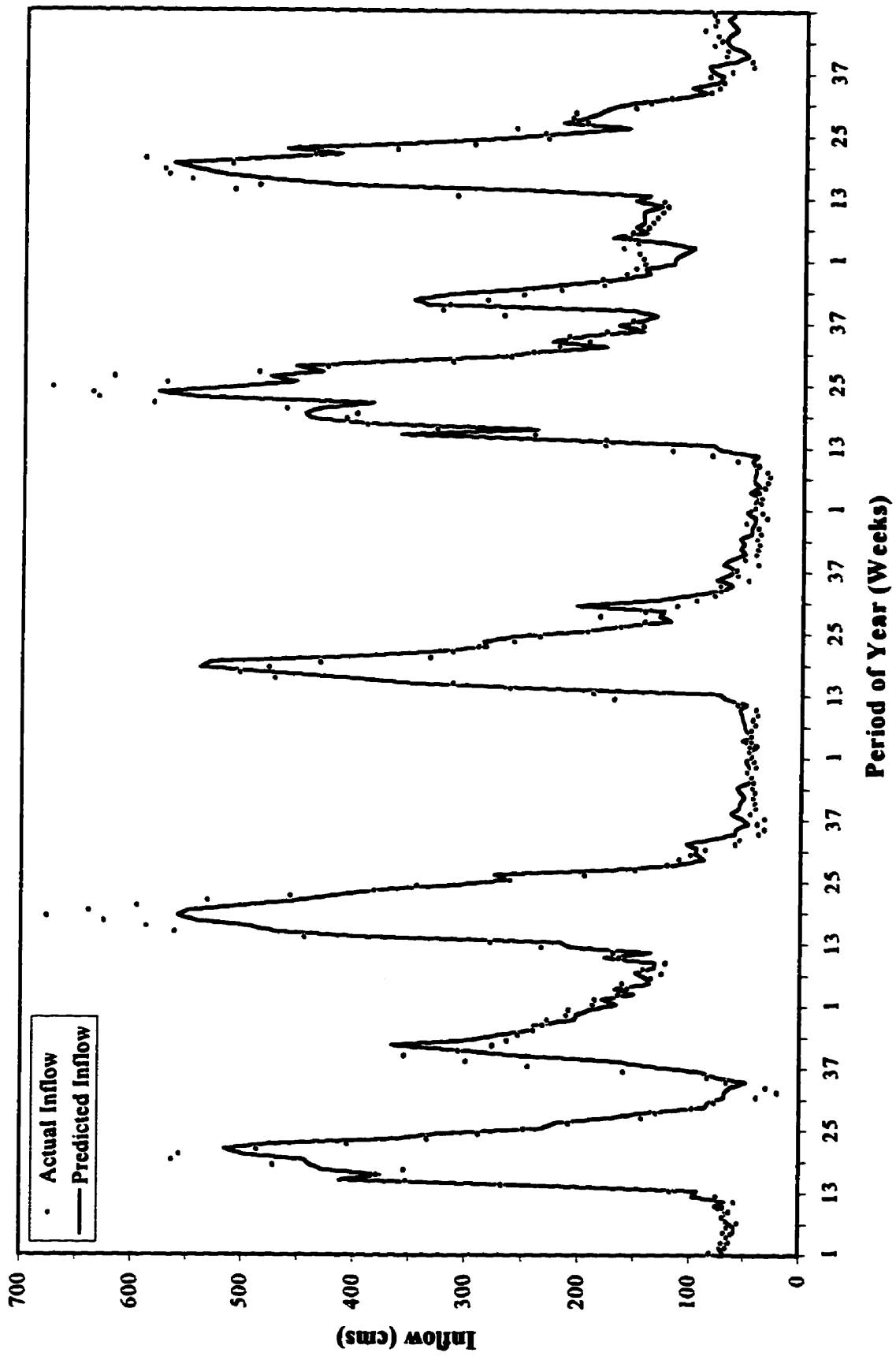


Figure C-b-1. Namakan Lake Training Results, 1965-69, two-week lead.

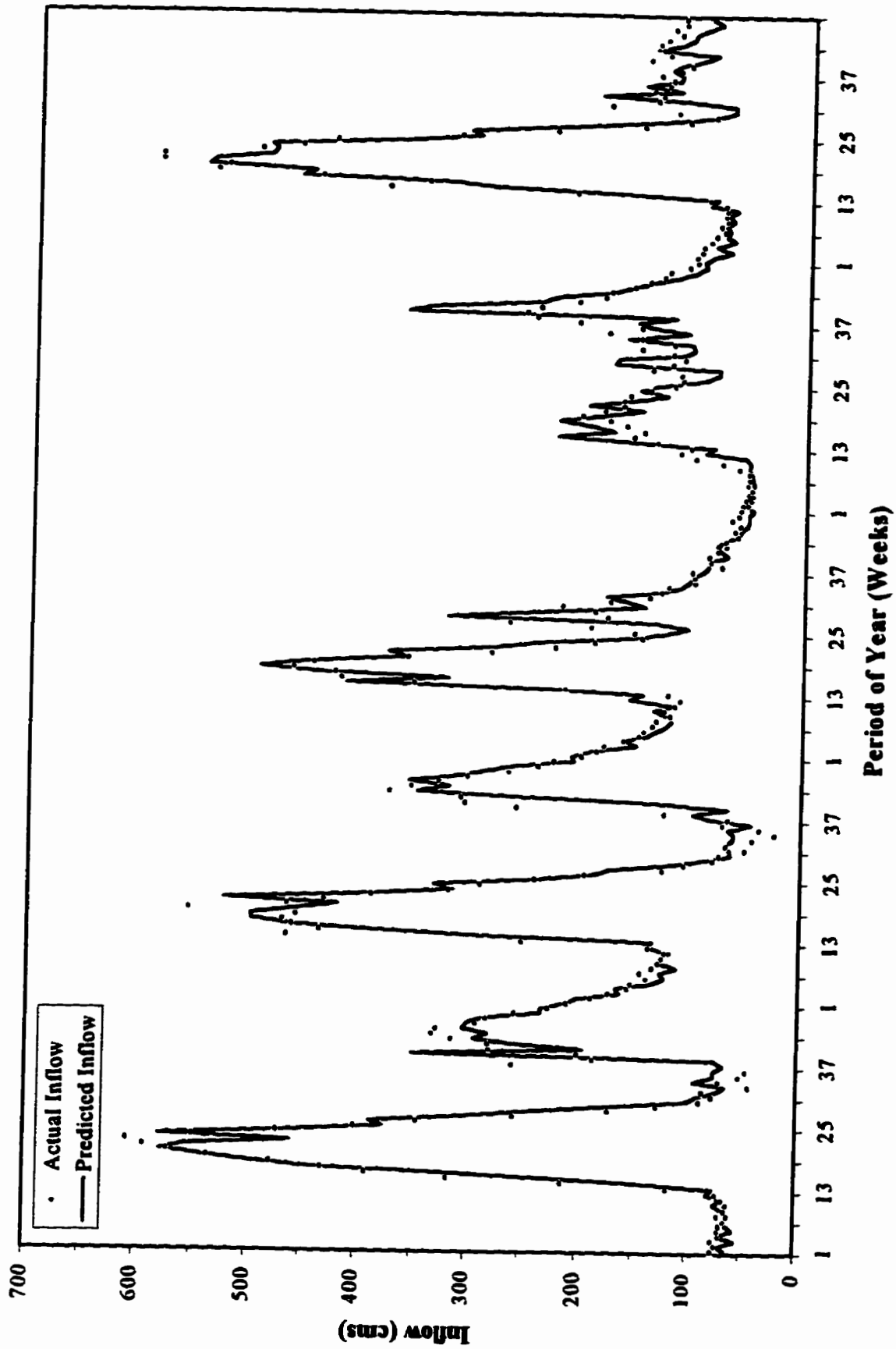


Figure C-b-2. Namakan Lake Training Results, 1970-74, two-week lead.

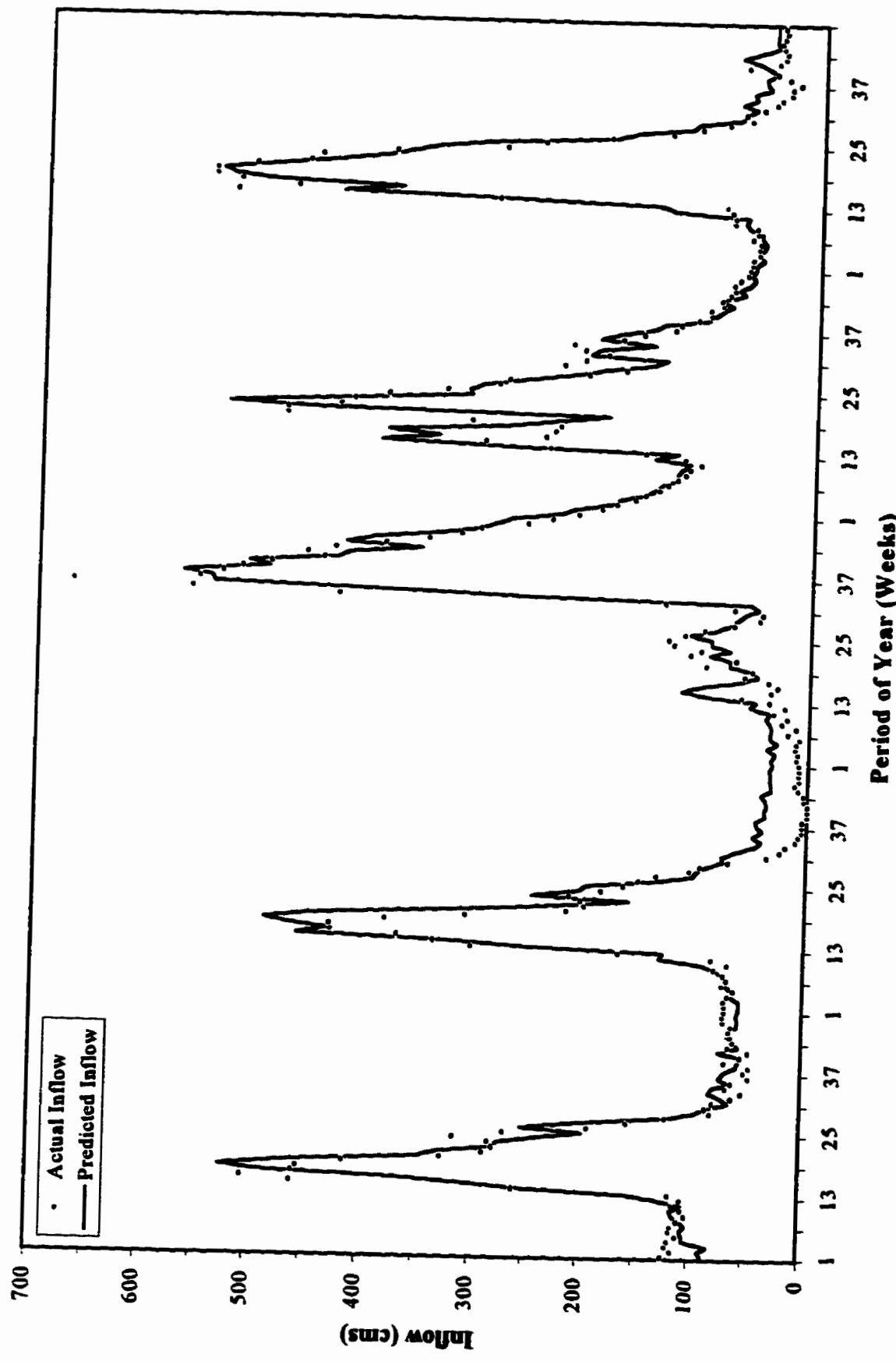


Figure C-b-3. Namakan Lake Training Results, 1975-79, two-week lead.

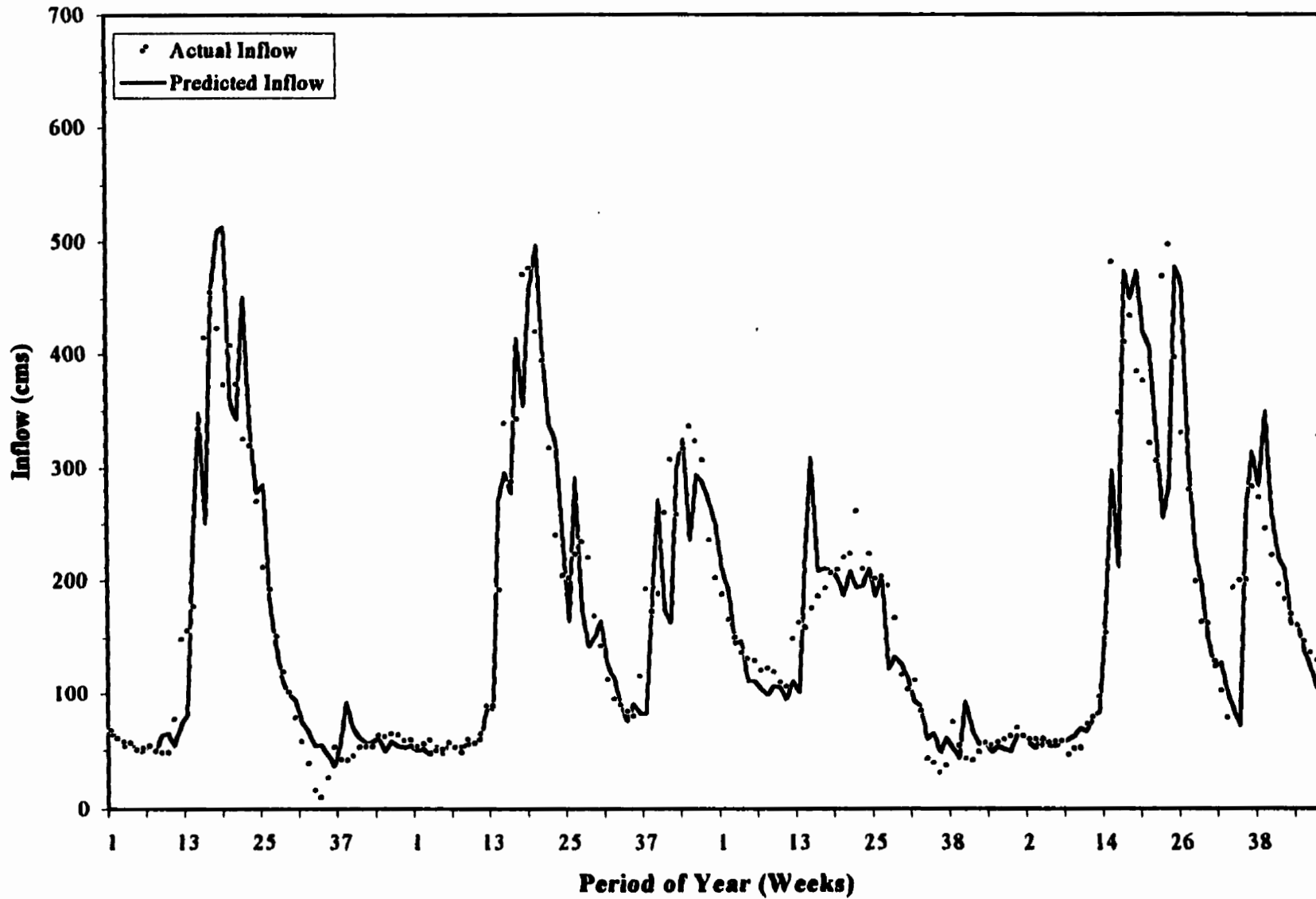


Figure C-b-4. Namakan Lake Training Results, 1980-85, two-week lead.

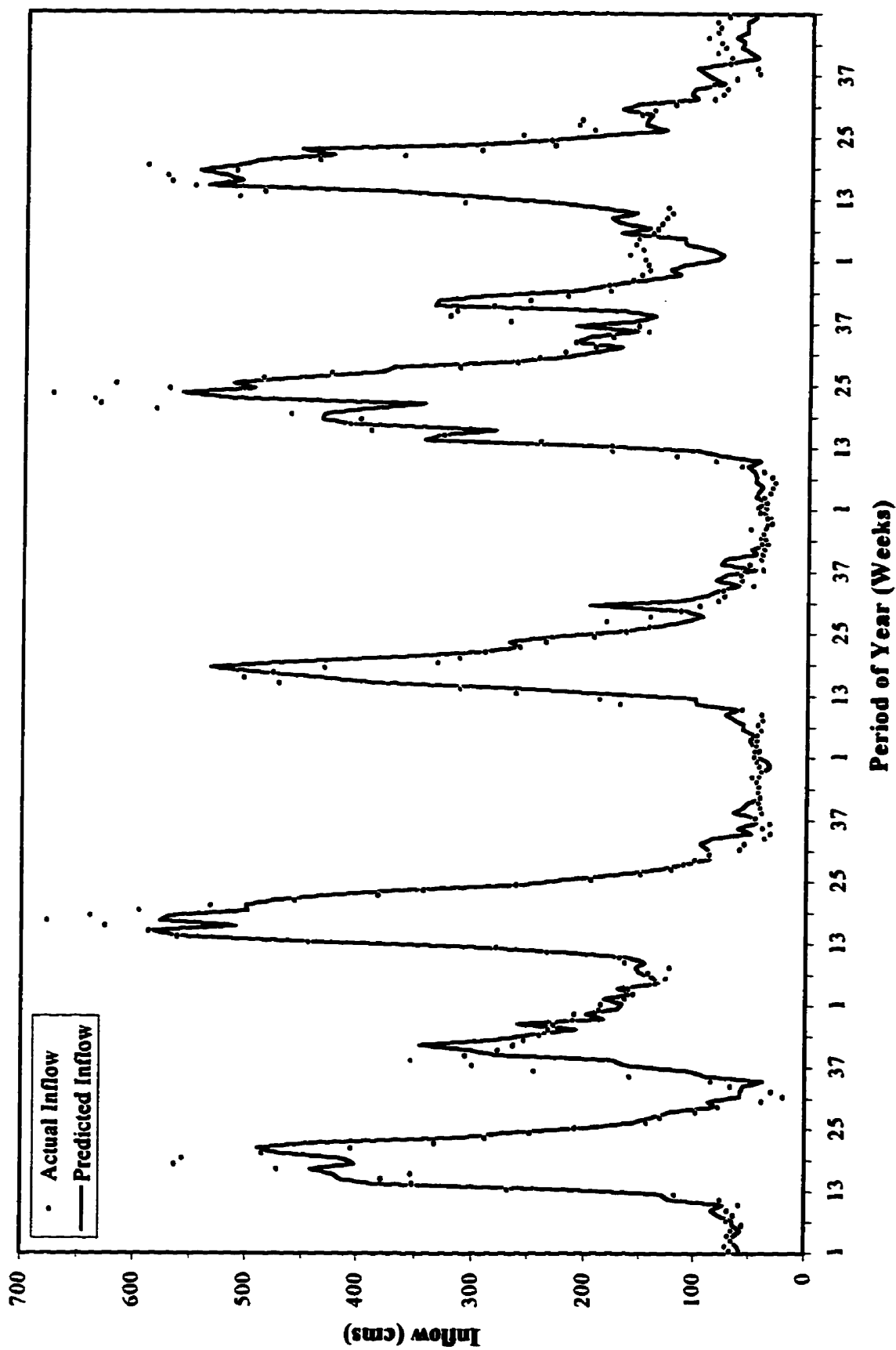


Figure C-c-1. Namakan Lake Training Results, 1965-69, three-week lead.

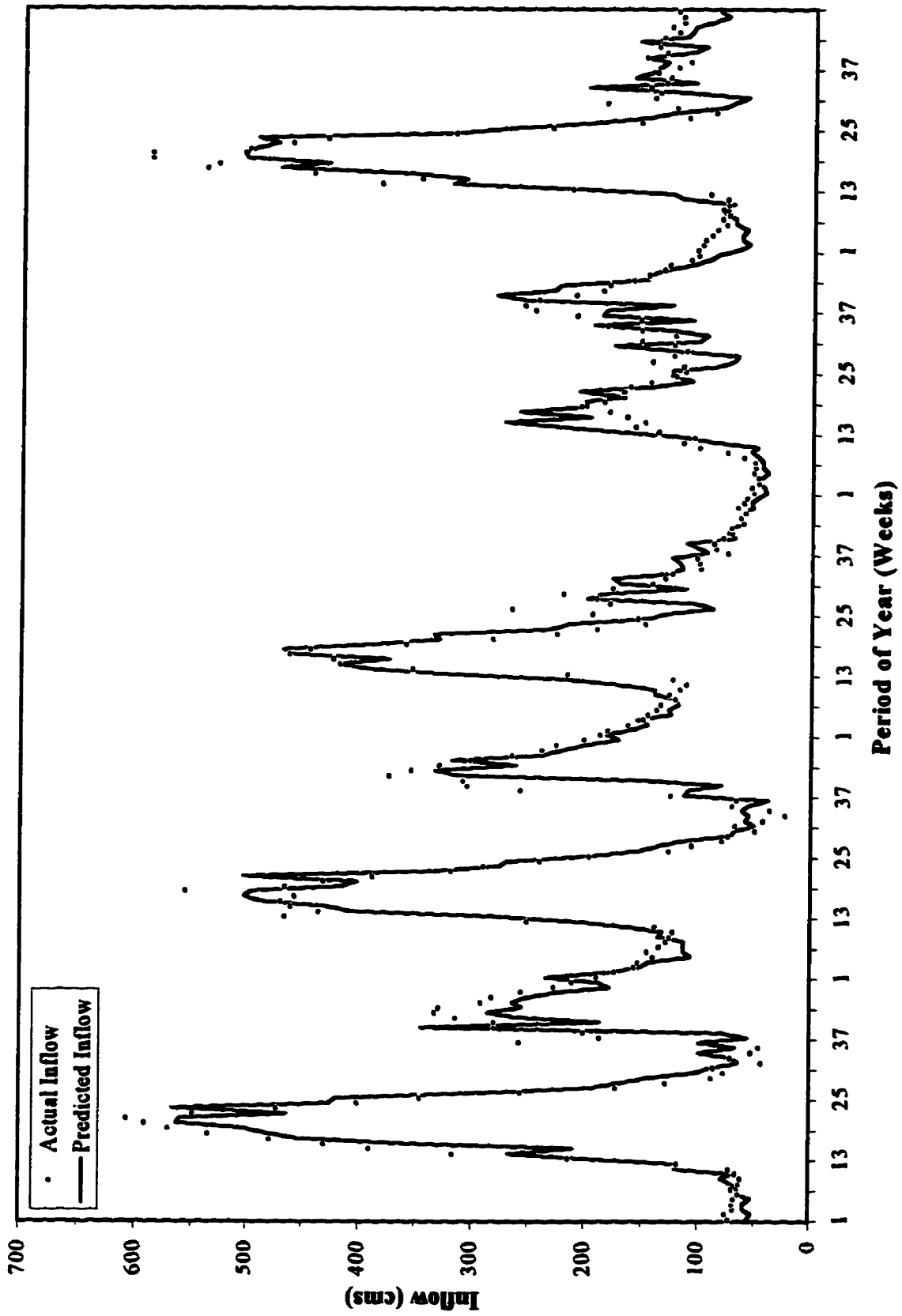


Figure C-c-2. Namakan Lake Training Results, 1970-74, three-week lead.



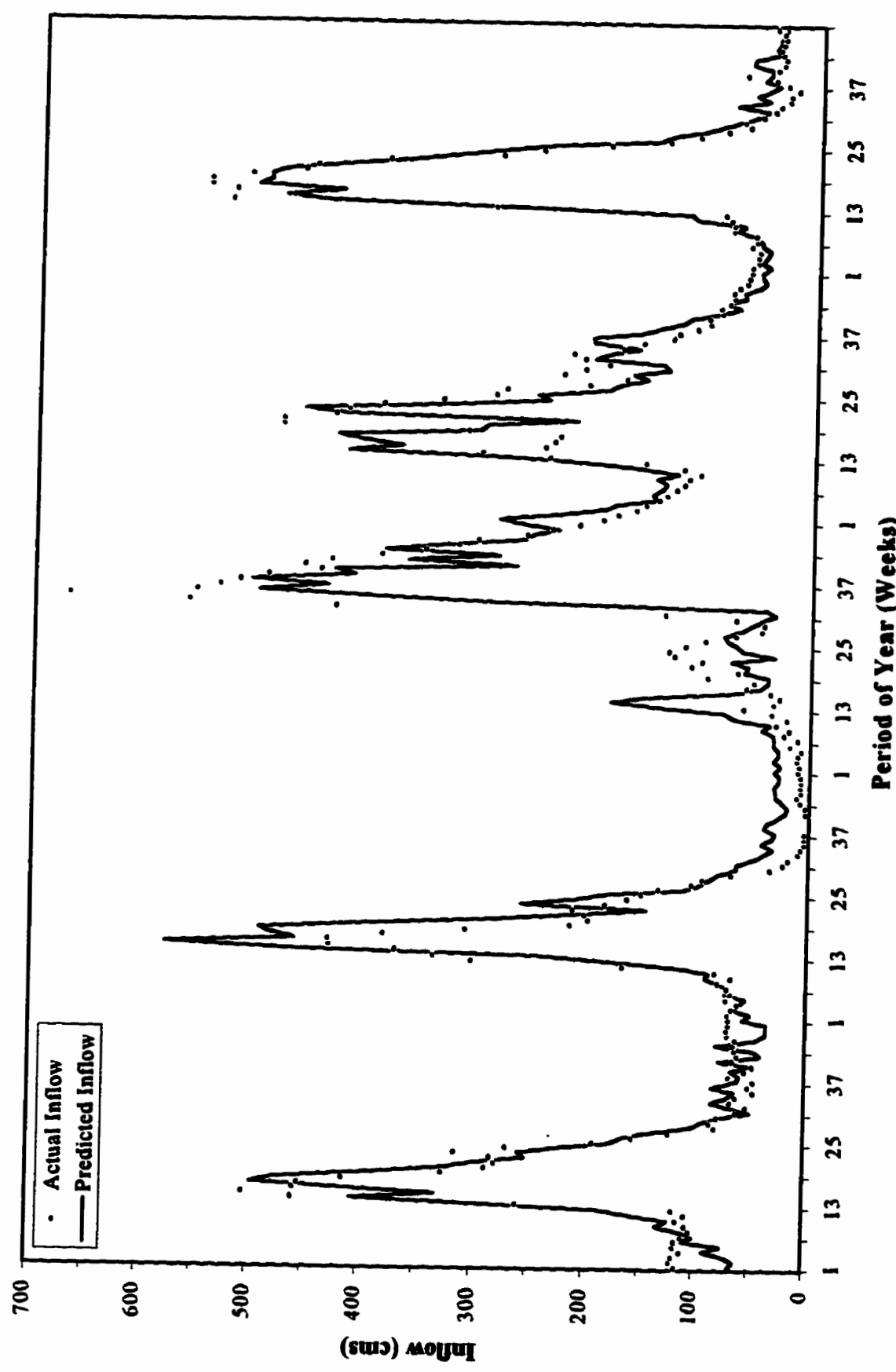


Figure C-c-3. Namakan Lake Training Results, 1975-79, three-week lead.

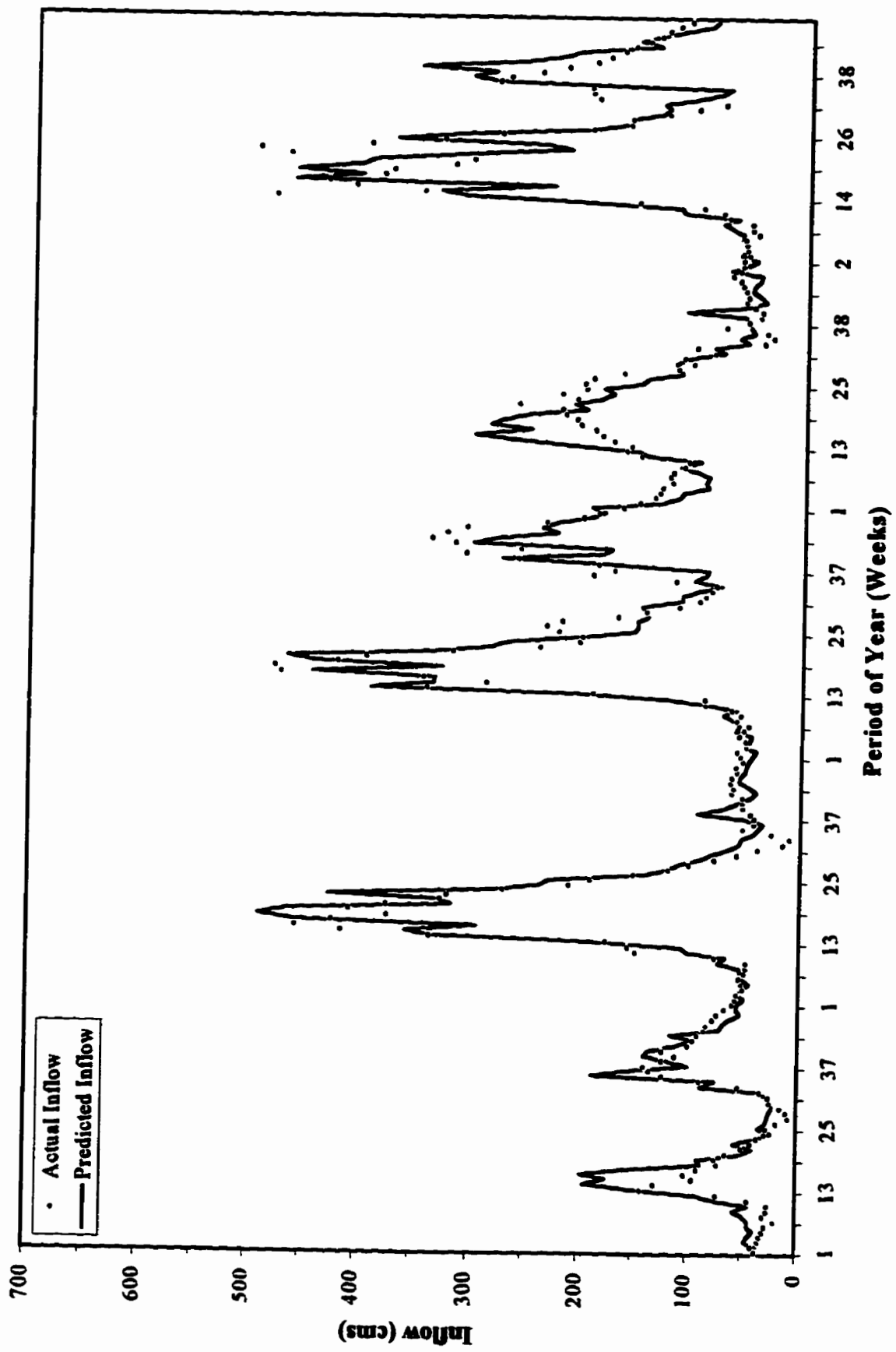


Figure C-c-4. Namakan Lake Training Results, 1980-85, three-week lead.

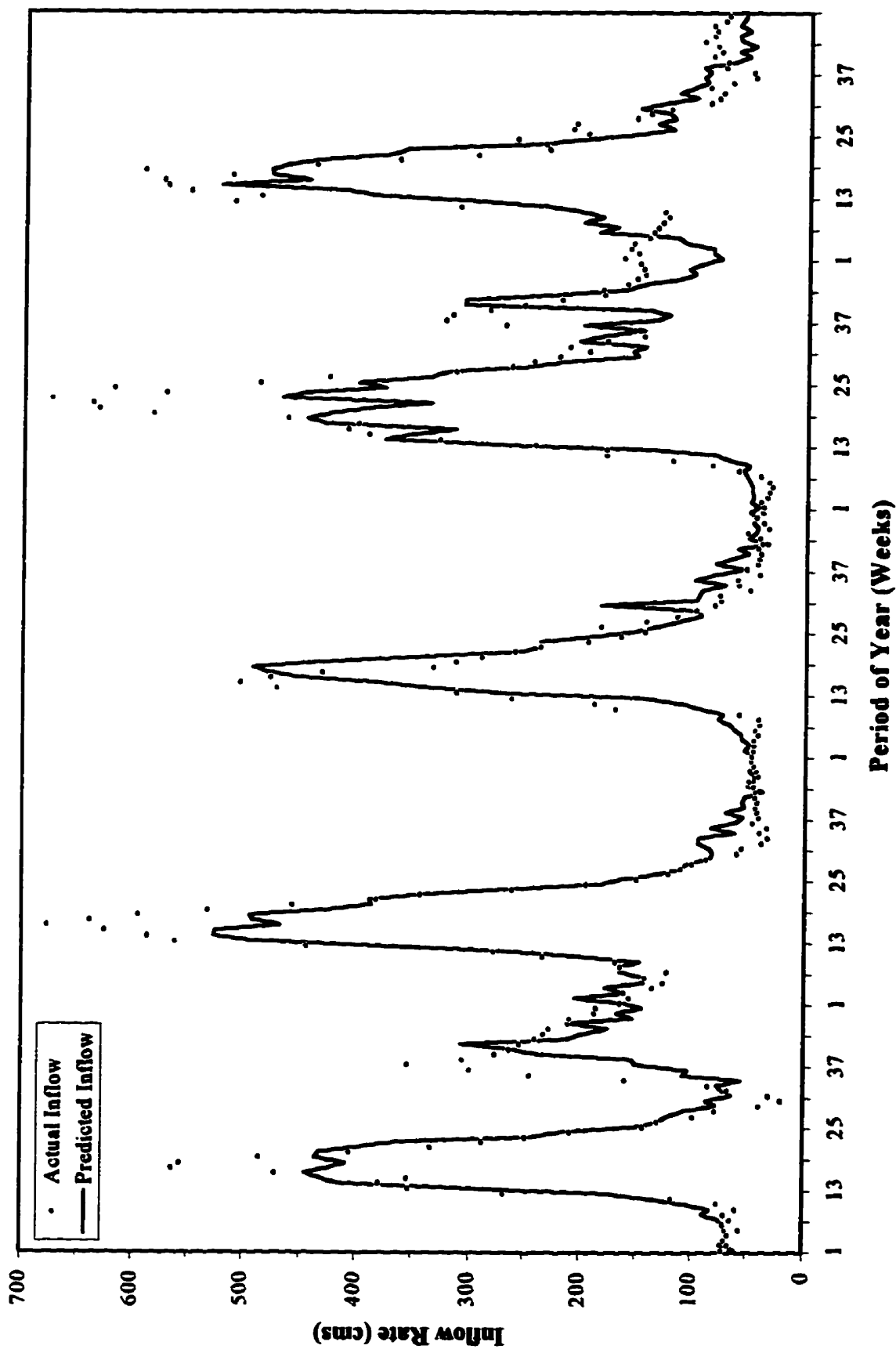
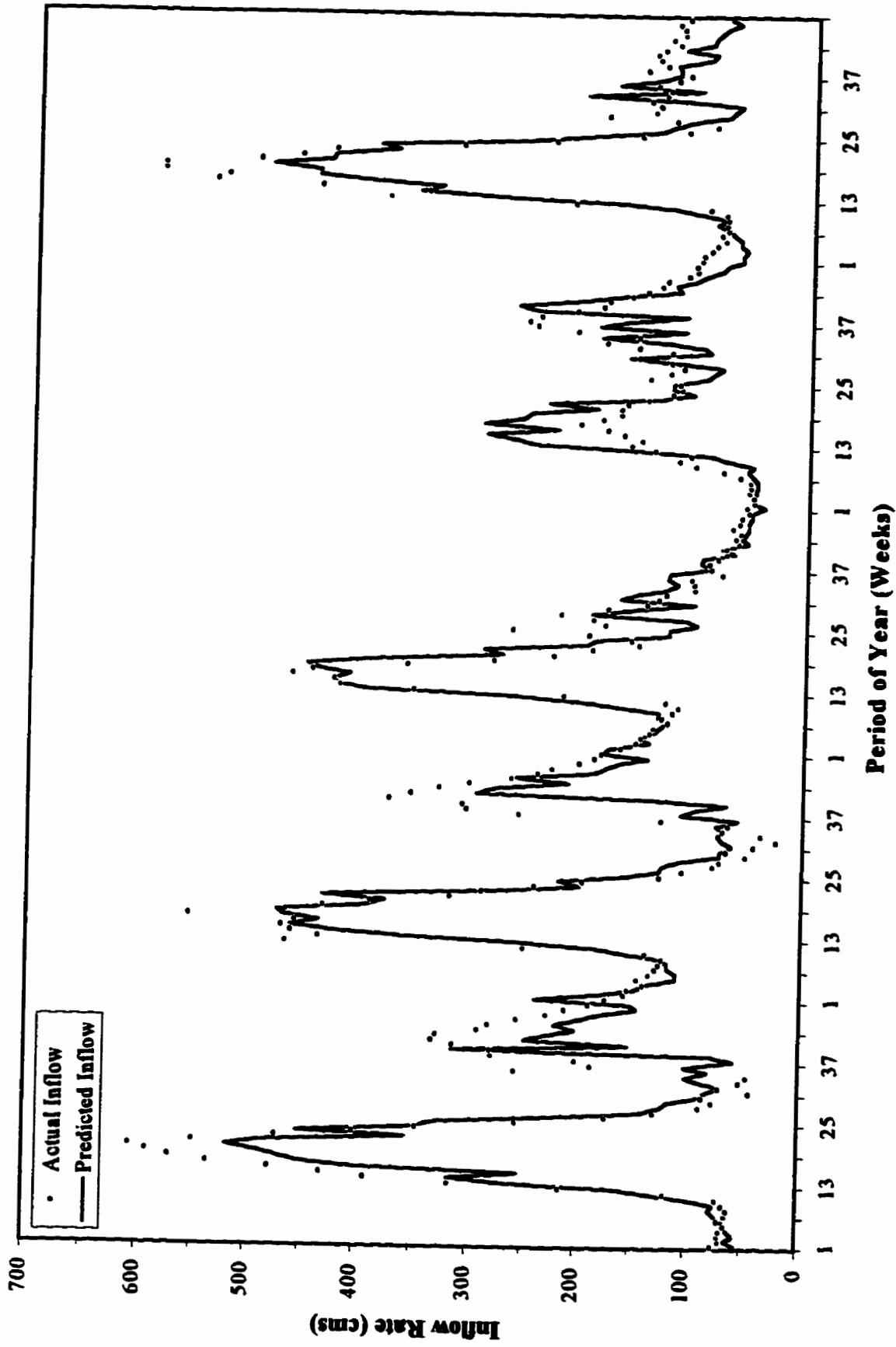


Figure C-d-1. Namakan Lake Training Results, 1965-69, four-week lead.



**Figure C-d-2. Namakan Lake Training Results, 1970-74, four-week lead.**

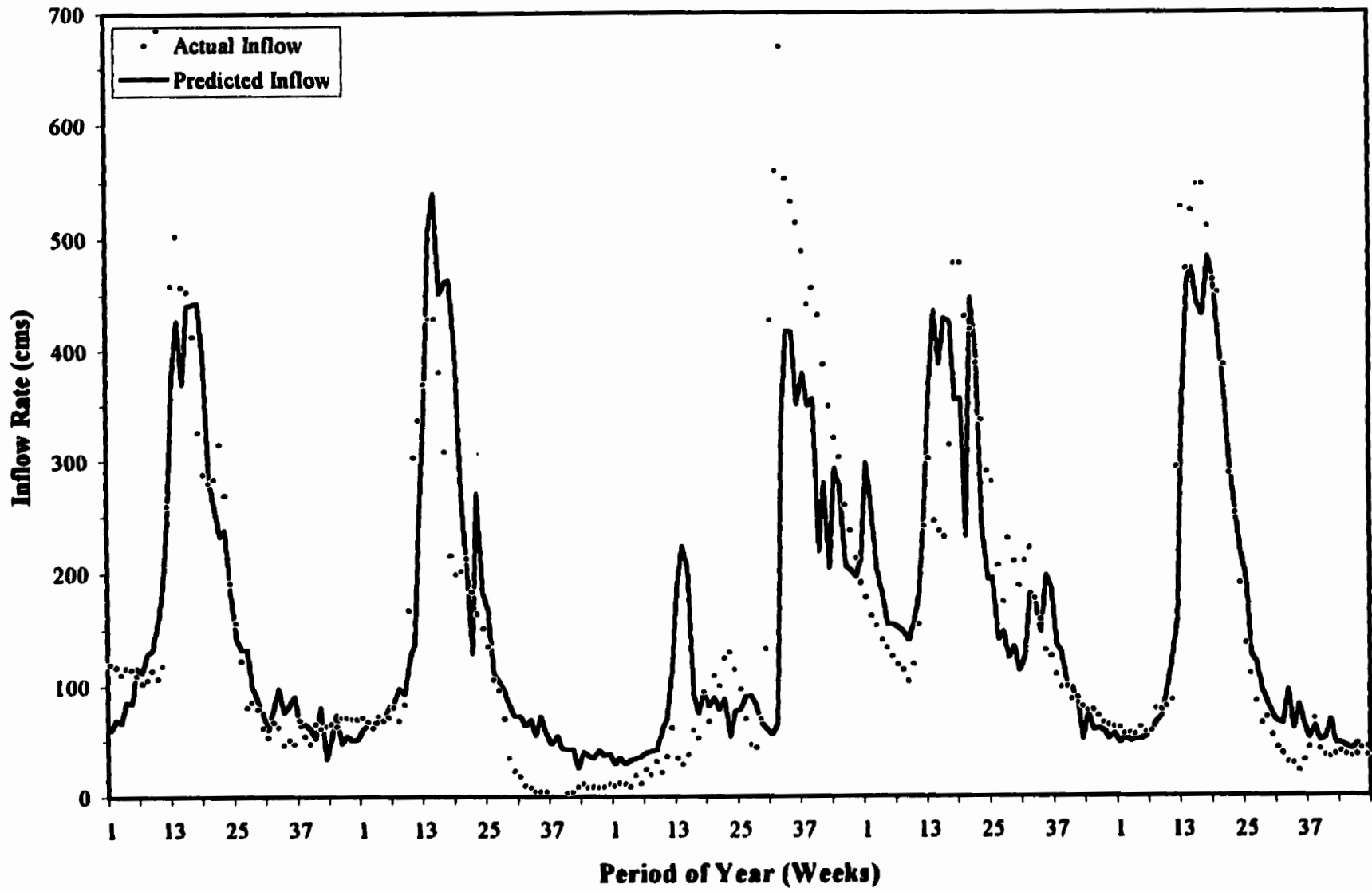
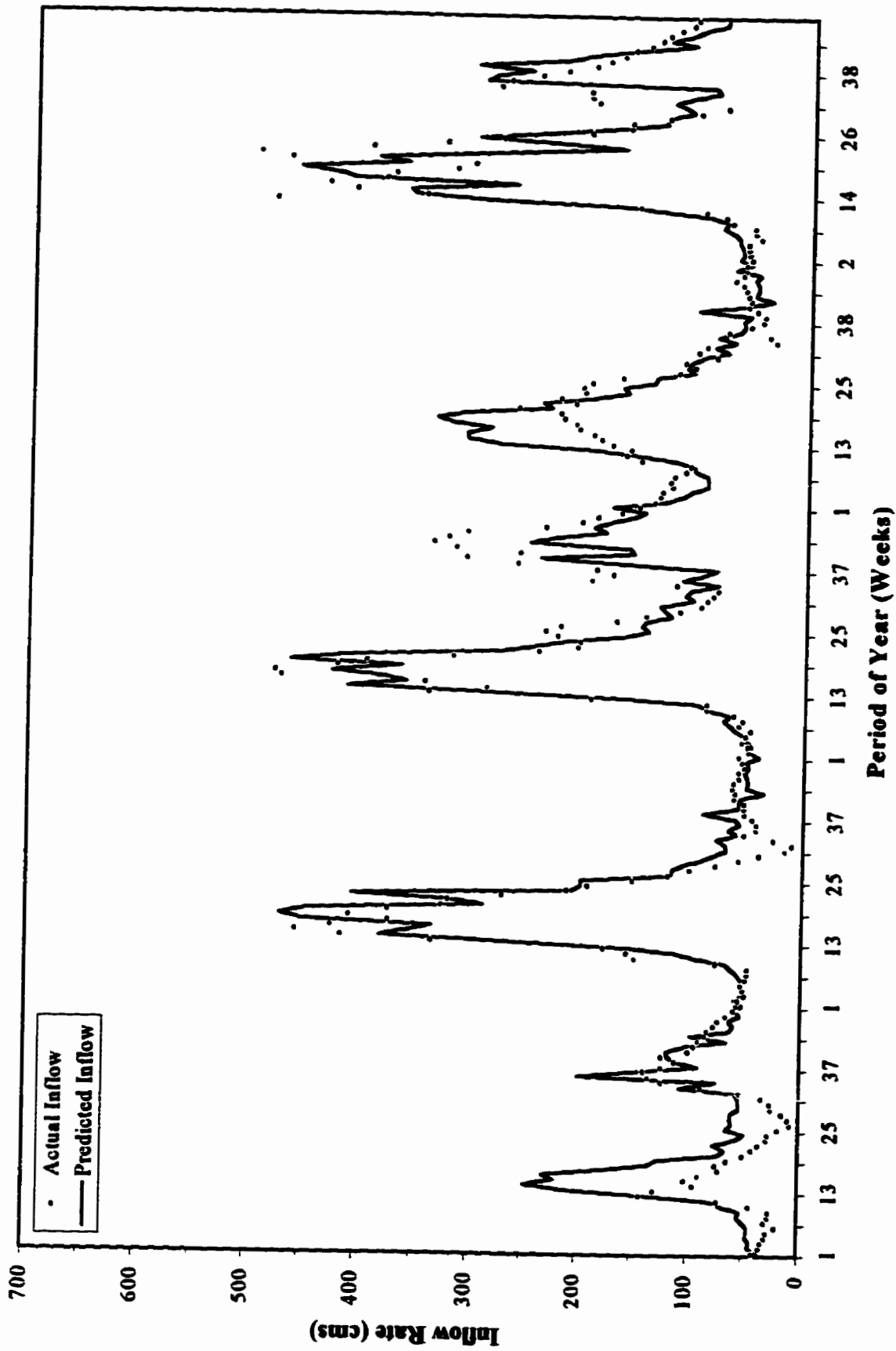
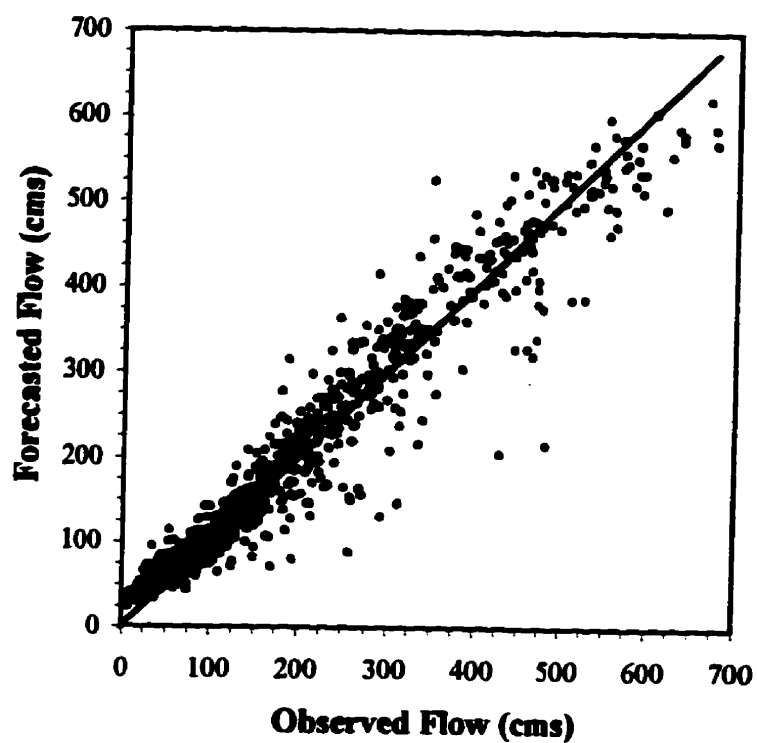


Figure C-d-3. Namakan Lake Training Results, 1975-79, four-week lead.

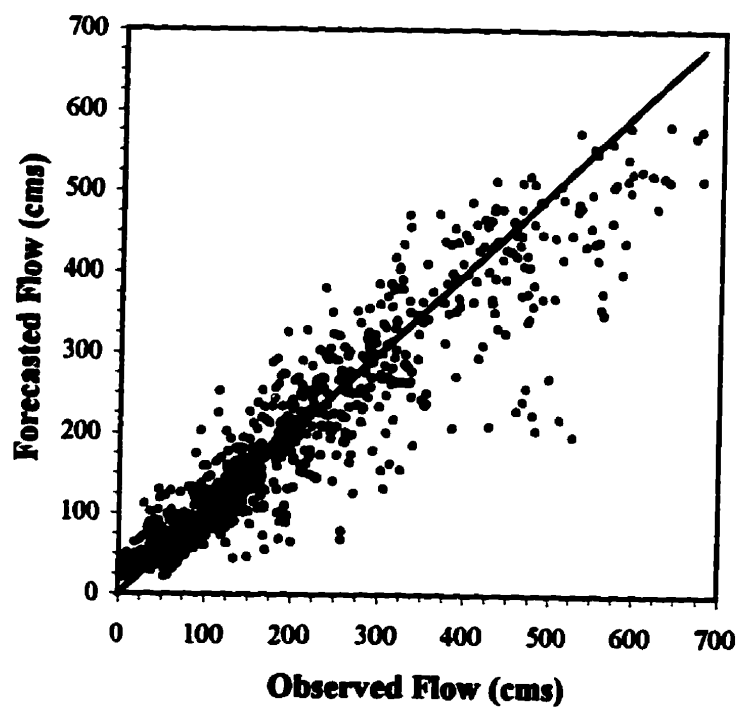


**Figure C-d-4. Namakan Lake Training Results, 1980-85, four-week lead.**

**APPENDIX D:****NAMAKAN LAKE SCATTER PLOTS FOR TRAINING SETS****D-1: EXPERIMENT #1****D-a-1: One-Week Lead****D-a-2: Two-Week Lead****D-a-3: Three-Week Lead****D-a-4: Four-Week Lead****D-2: EXPERIMENT #2****D-b-1: One-Week Lead****D-b-2: Two-Week Lead****D-b-3: Three-Week Lead****D-b-4: Four-Week Lead**

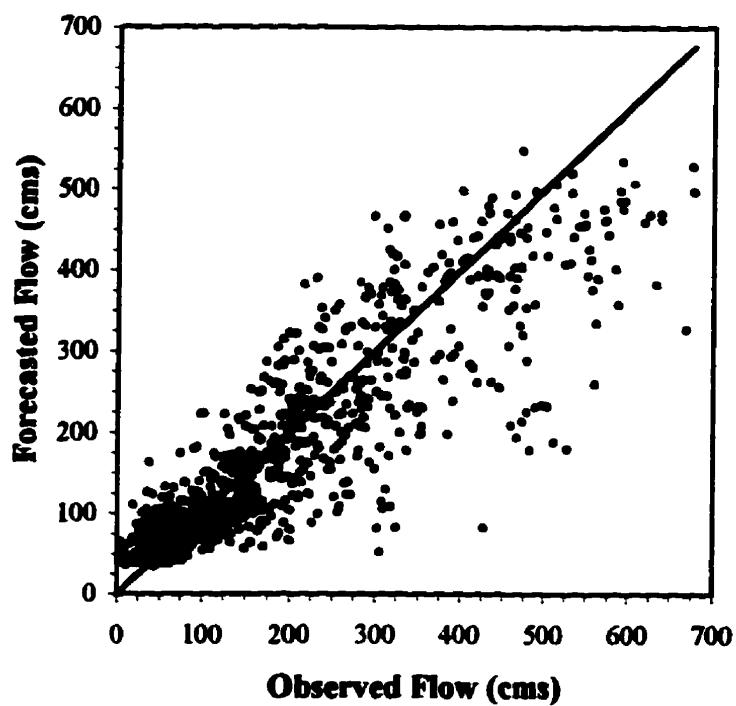


**FigureD-a-1. Observed versus Best 1-week Forecast of local inflow to Namakan Lake for 1965-85 using Exp. #1 ANN.**

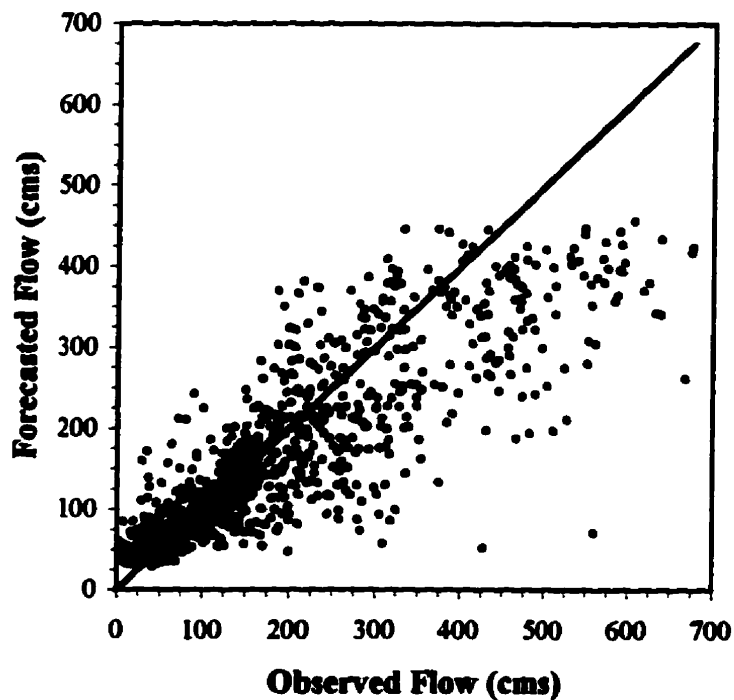


**Figure D-a-2. Observed versus Best 2-week Forecast of local inflow to Namakan Lake for 1965-85 using Exp. #1 ANN.**

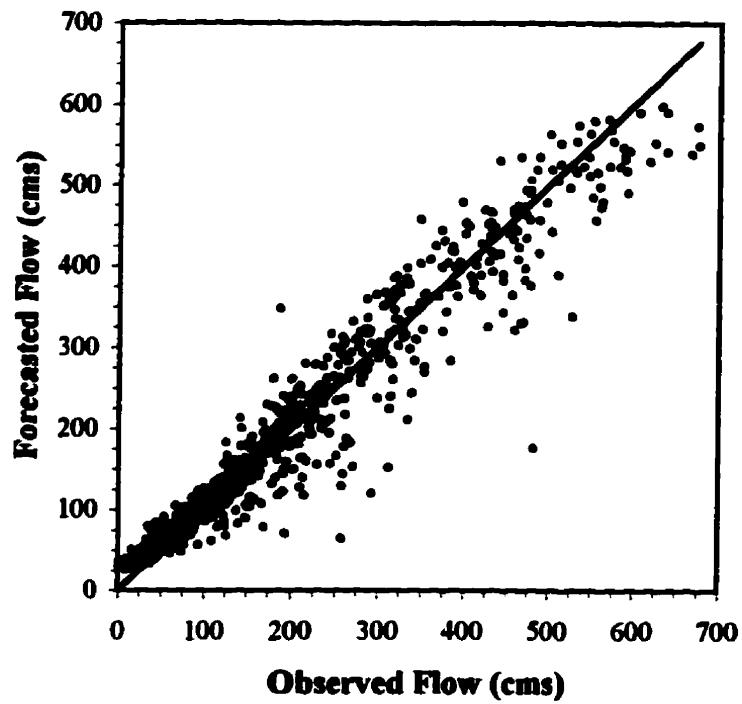




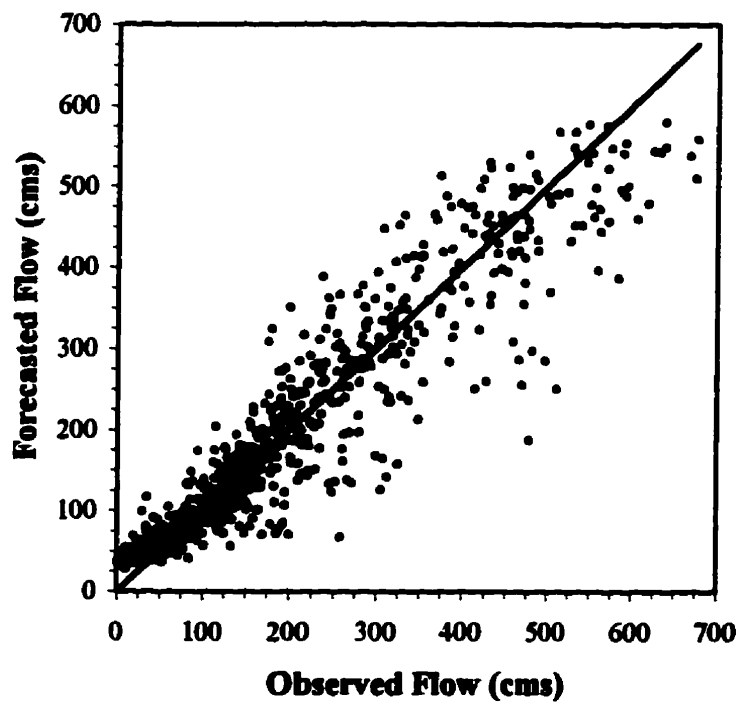
**Figure D-a-3. Observed versus Best 3-week Forecast of local inflow to Namakan Lake for 1965-85 using Exp. #1 ANN.**



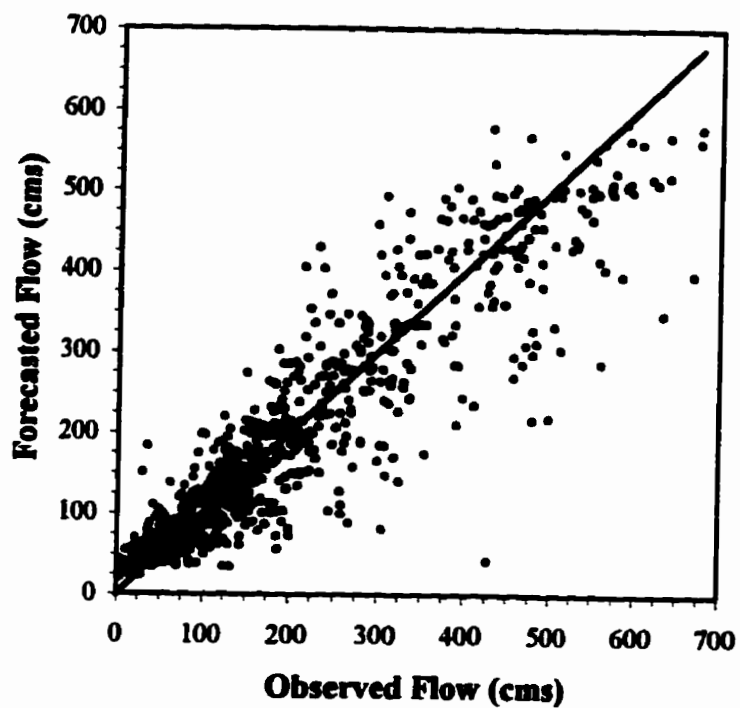
**Figure D-a-4. Observed versus Best 4-week Forecast of local inflow to Namakan Lake for 1965-85 using Exp. #1 ANN.**



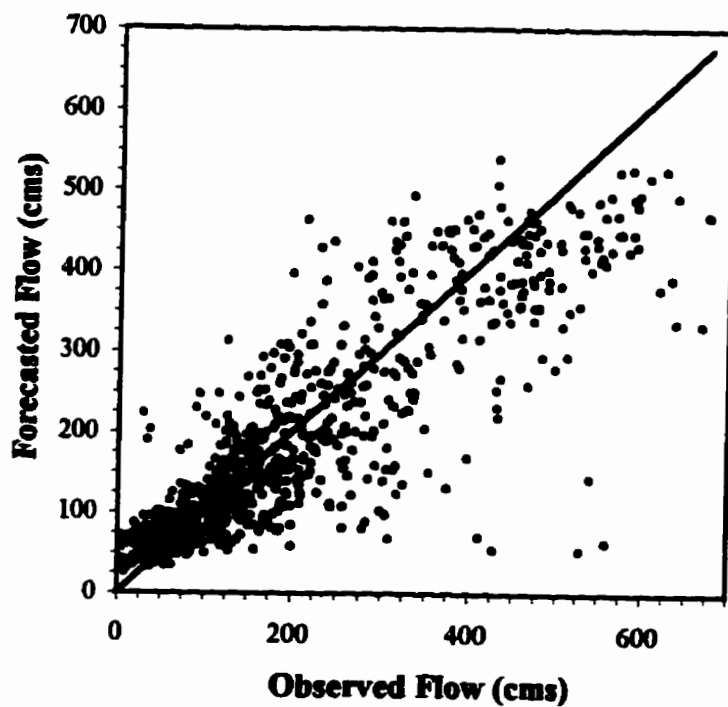
**Figure D-b-1. Observed versus Best 1-week Forecast of local inflow to Namakan Lake for 1965-85 using Exp. #2 ANN.**



**Figure D-b-2. Observed versus Best 2-week Forecast of local inflow to Namakan Lake for 1965-85 using Exp. #2 ANN.**



**Figure D-b-3. Observed versus Best 3-week Forecast of local inflow to Namakan Lake for 1965-85 using Exp. #2 ANN.**



**Figure D-b-4. Observed versus Best 4-week Forecast of local inflow to Namakan Lake for 1965-85 using Exp. #2 ANN.**

**APPENDIX E:**

**RAINY LAKE LOCAL HISTORICAL DATA**

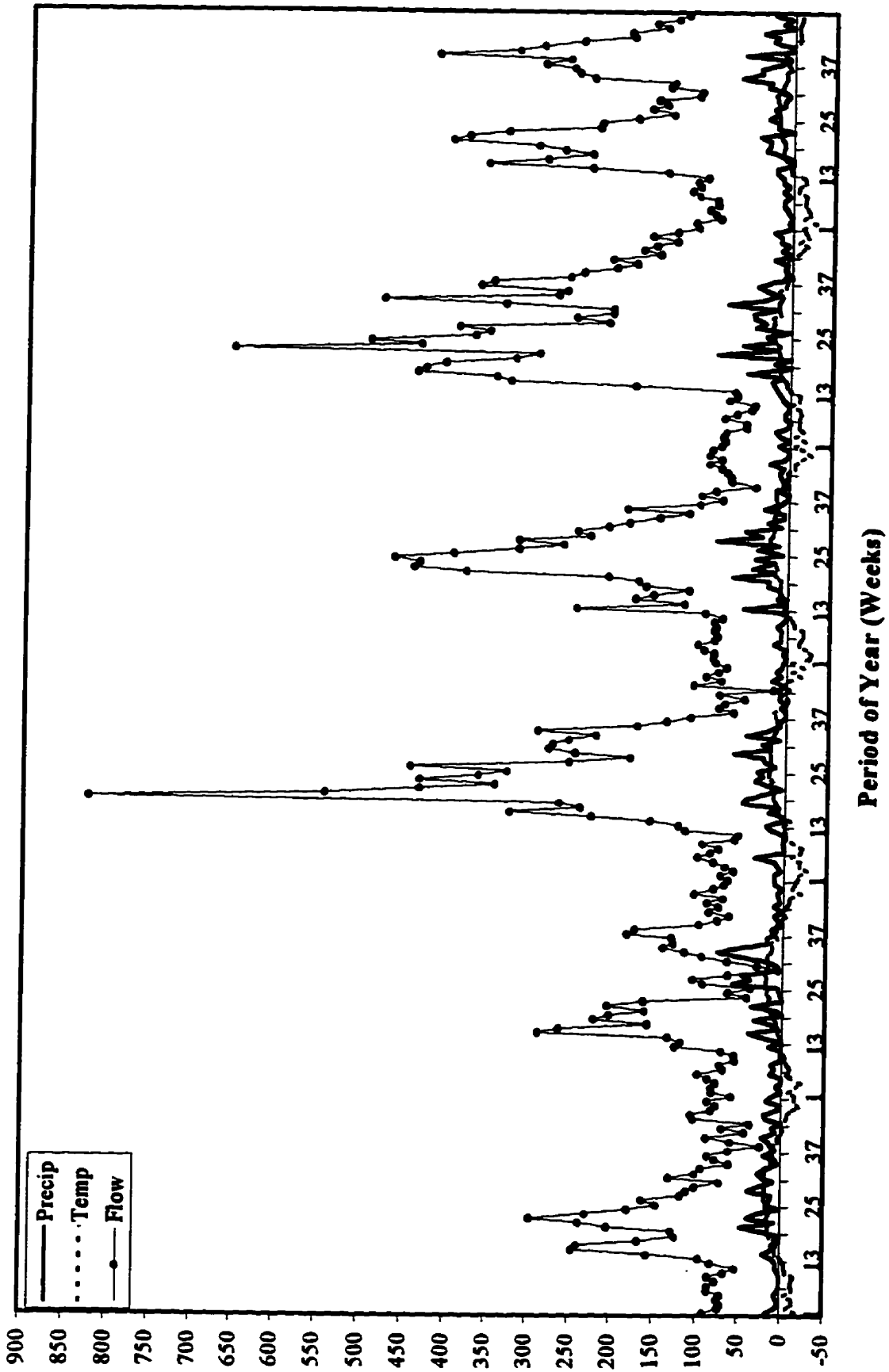


Figure E-1. Rainy Lake Historical Data, 1960-65.

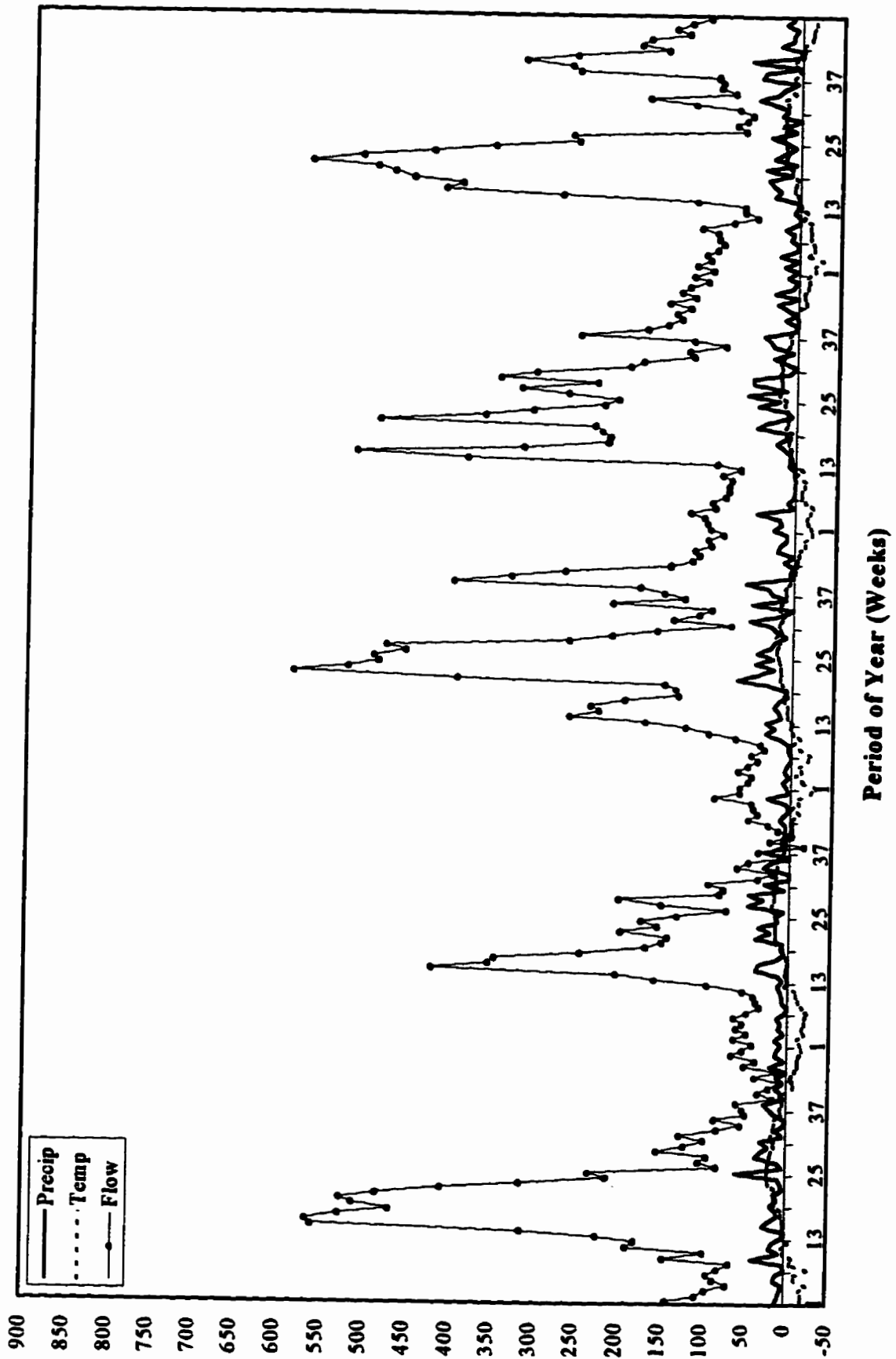


Figure E-2. Rainy Lake Historical Data, 1966-70.

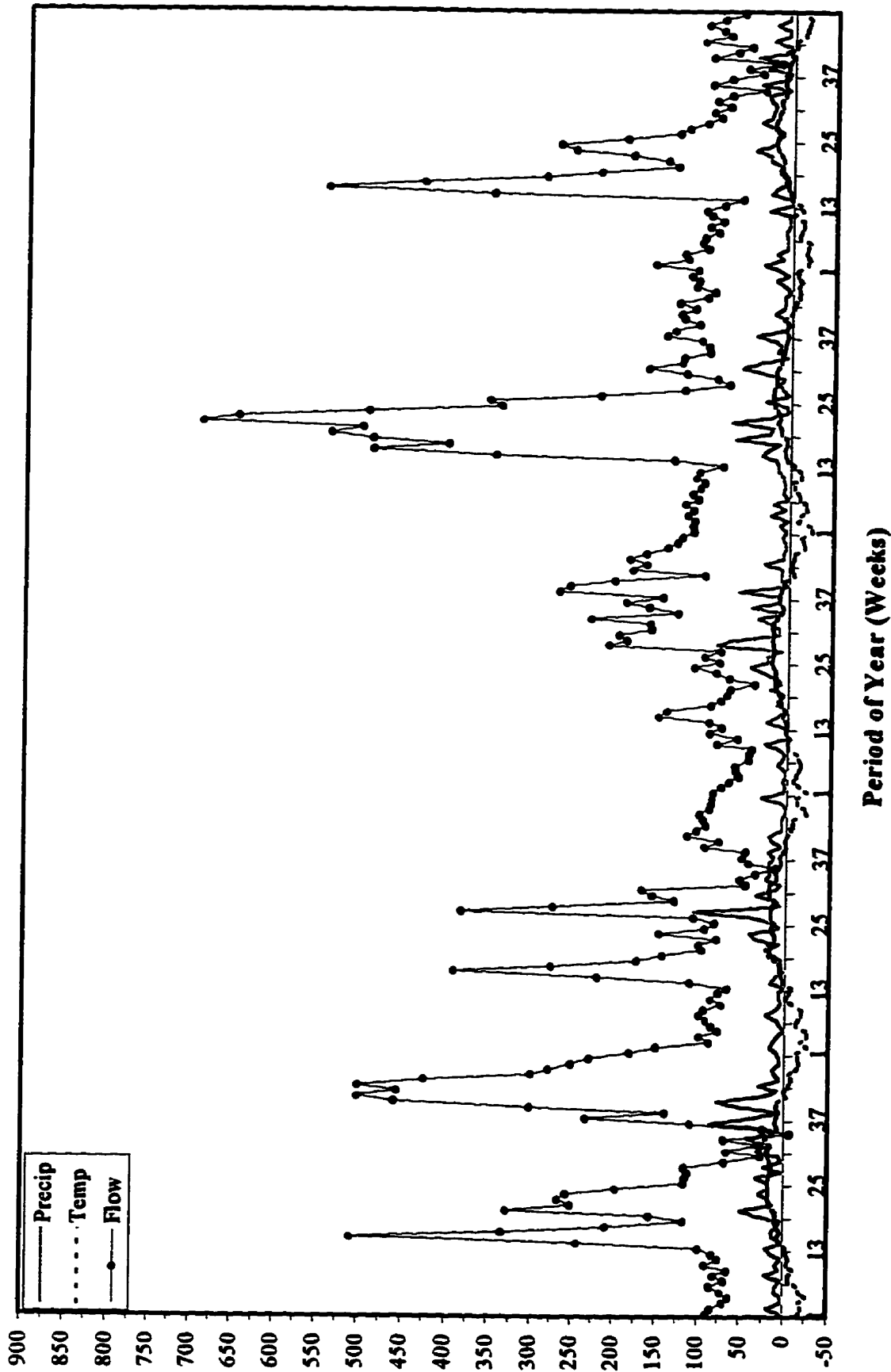


Figure E-3. Rainy Lake Historical Data, 1971-75.

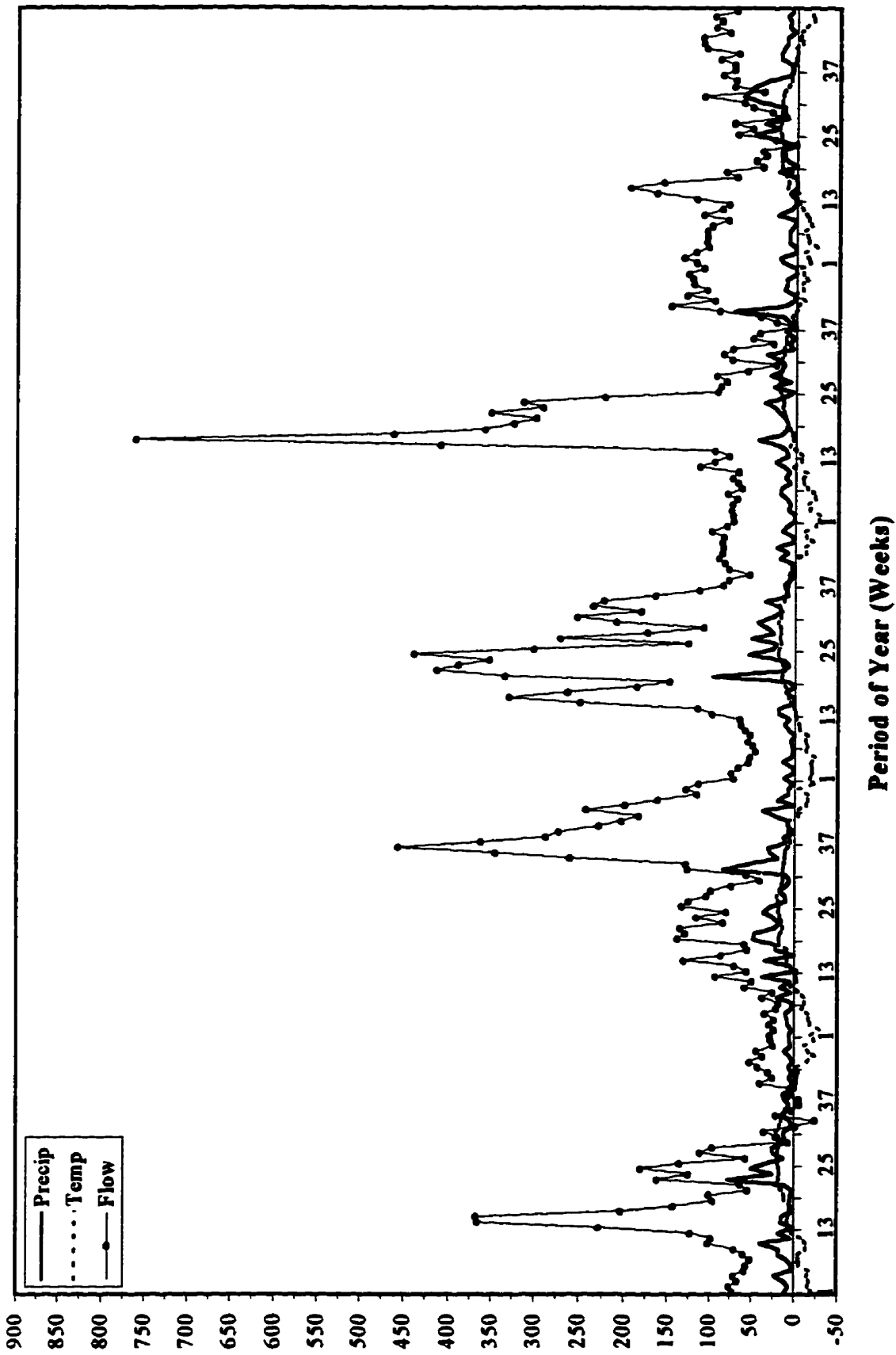


Figure E-4. Rainy Lake Historical Data, 1976-80.



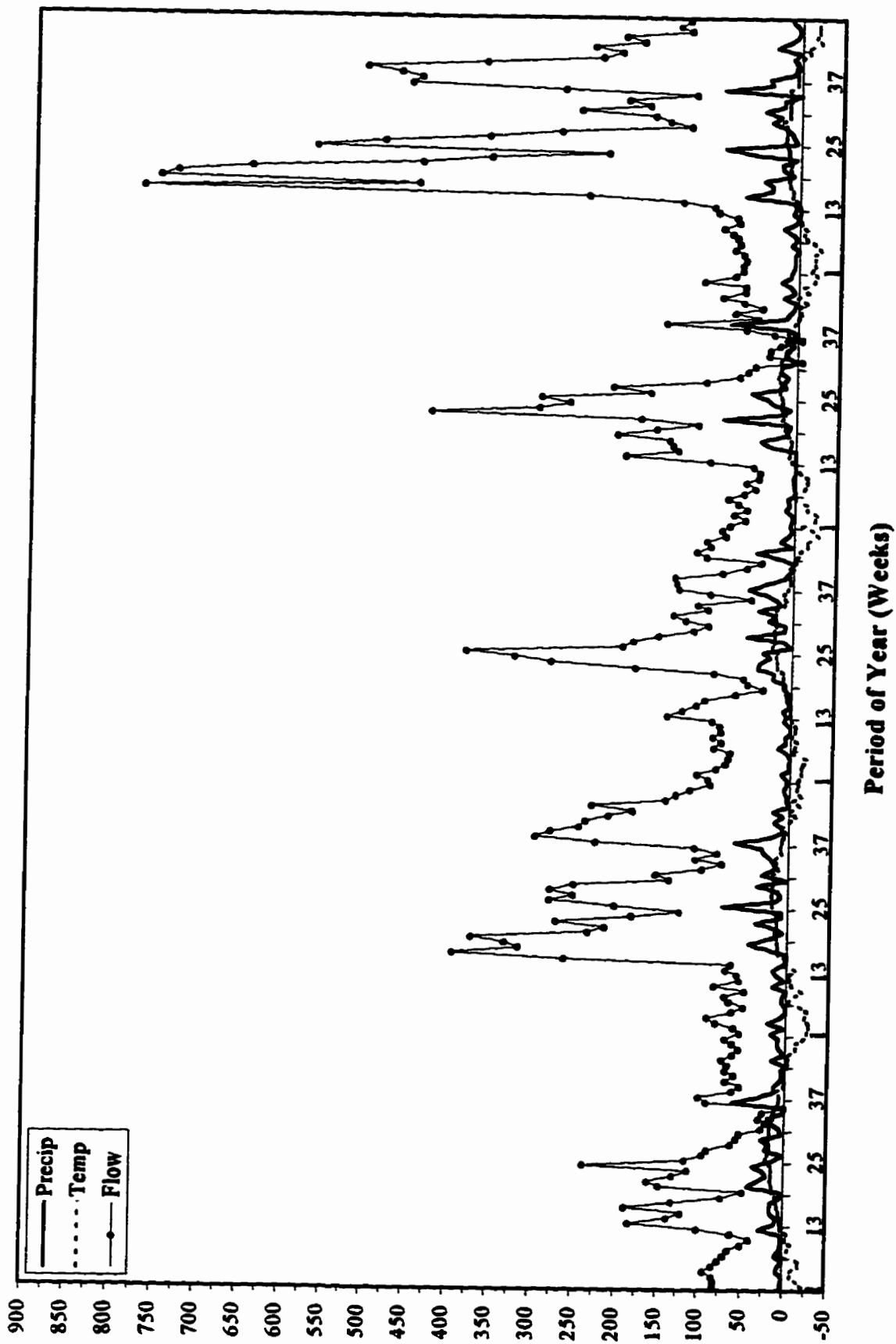
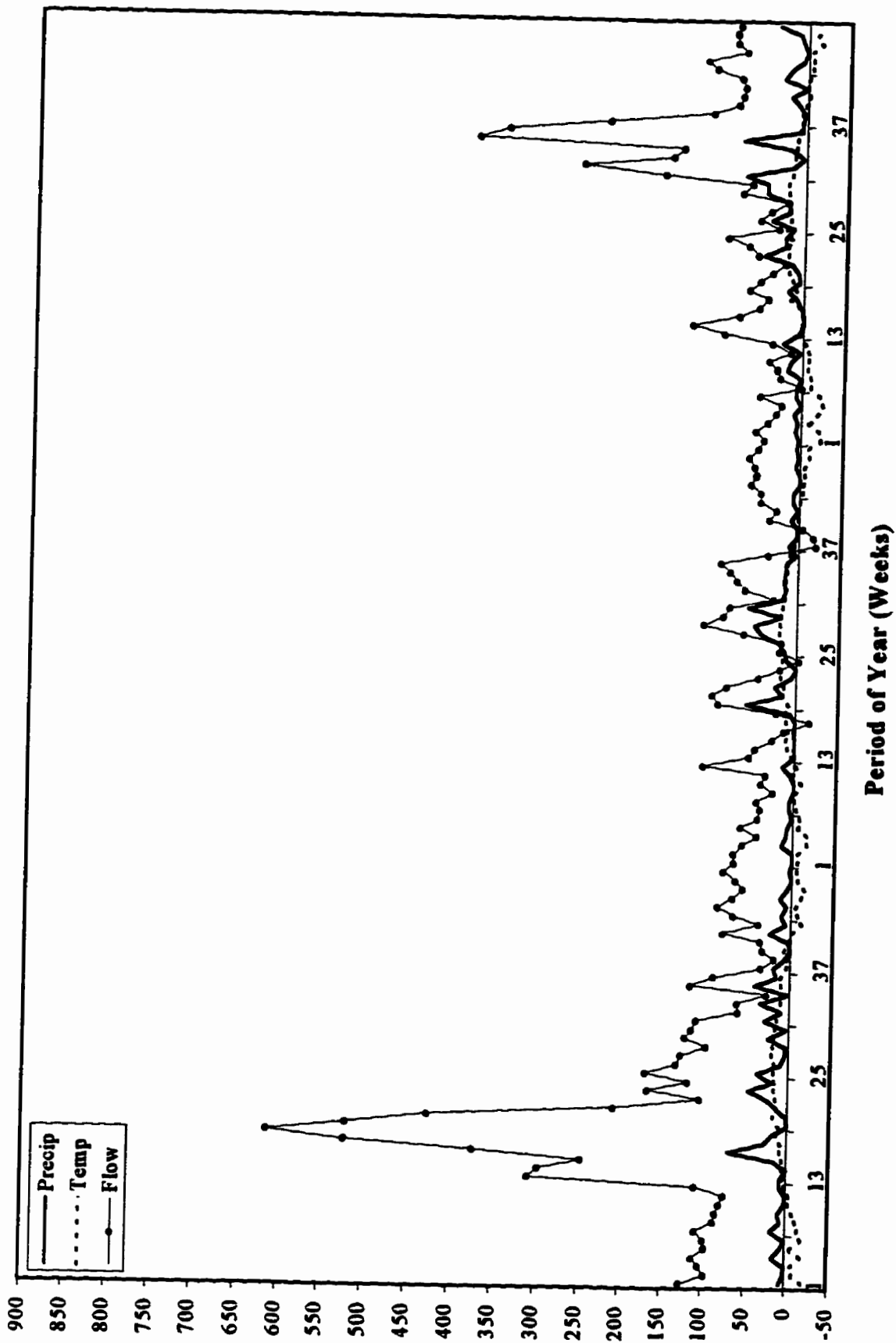


Figure E-5. Rainy Lake Historical Data, 1981-85.



**Figure E-6. Rainy Lake Historical Data, 1986-88.**

**APPENDIX F:**

**RAINY LAKE LOCAL TRAINING RESULTS**

**F-1: ONE-WEEK LEAD**

**F-a-1: 1965-1969**

**F-a-2: 1970-1974**

**F-a-3: 1975-1979**

**F-a-4: 1980-1985**

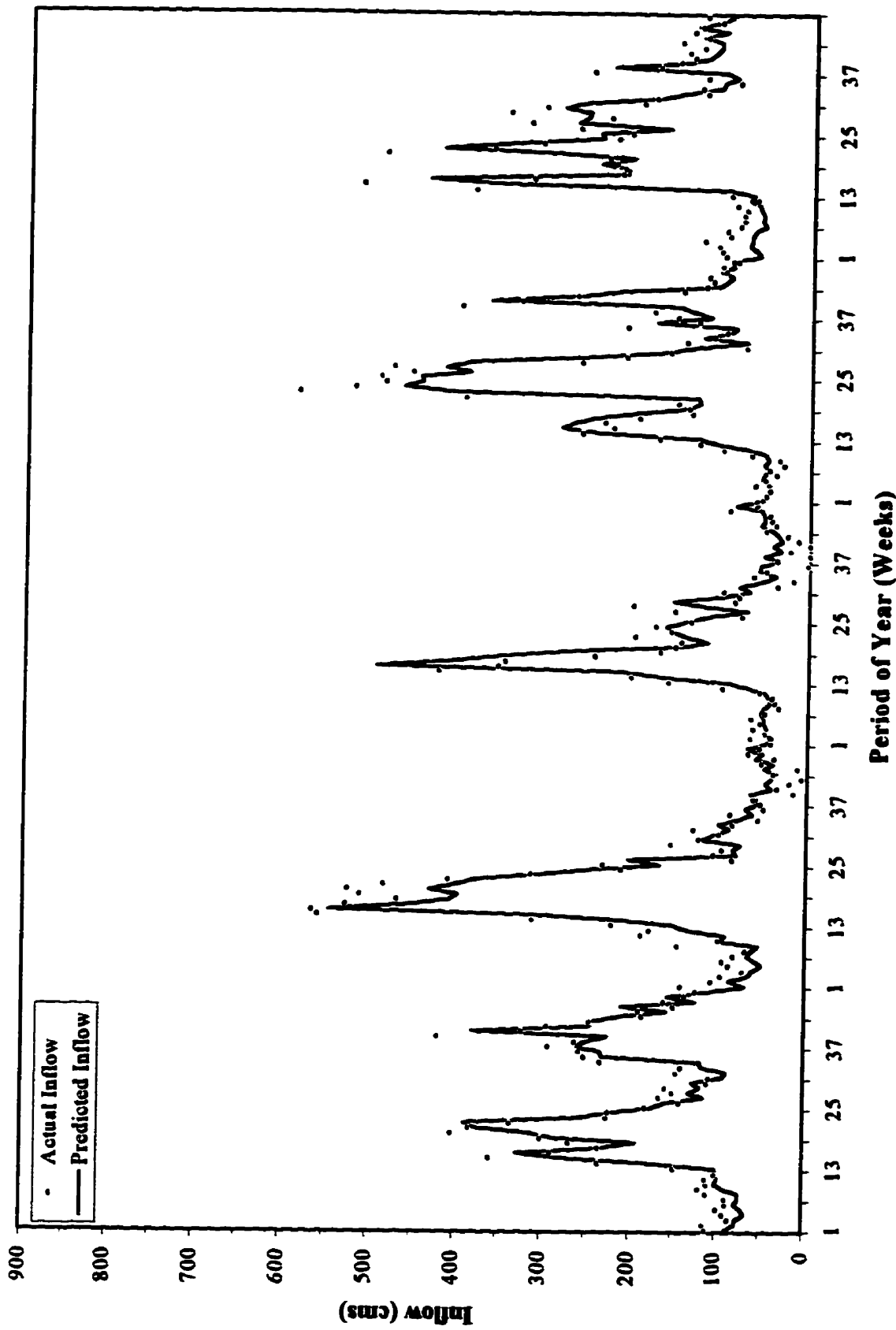
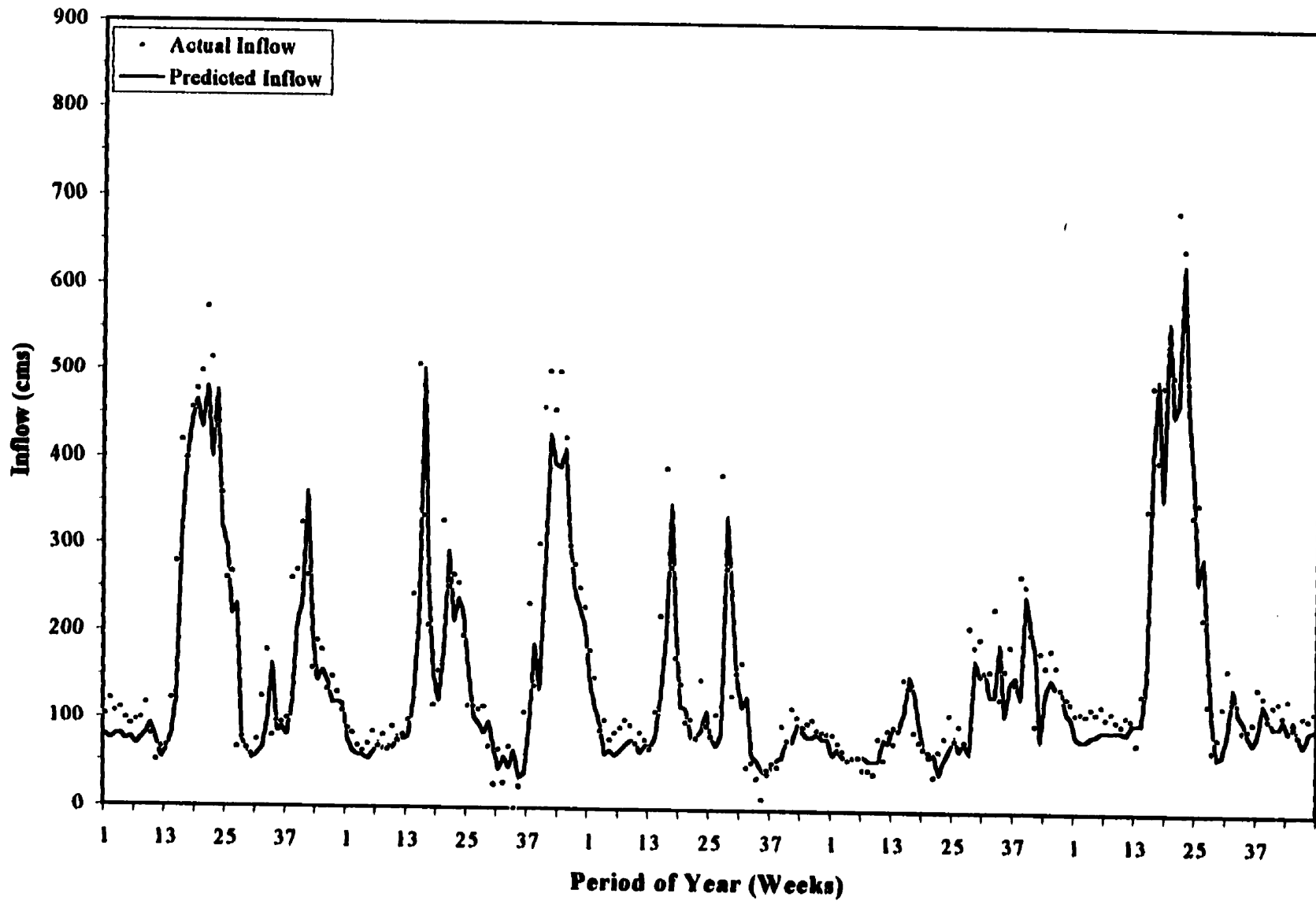


Figure F-a-1. Rainy Lake Training Results, 1965-69, one-week lead.



**Figure F-a-2. Rainy Lake Training Results, 1970-74, one-week lead.**

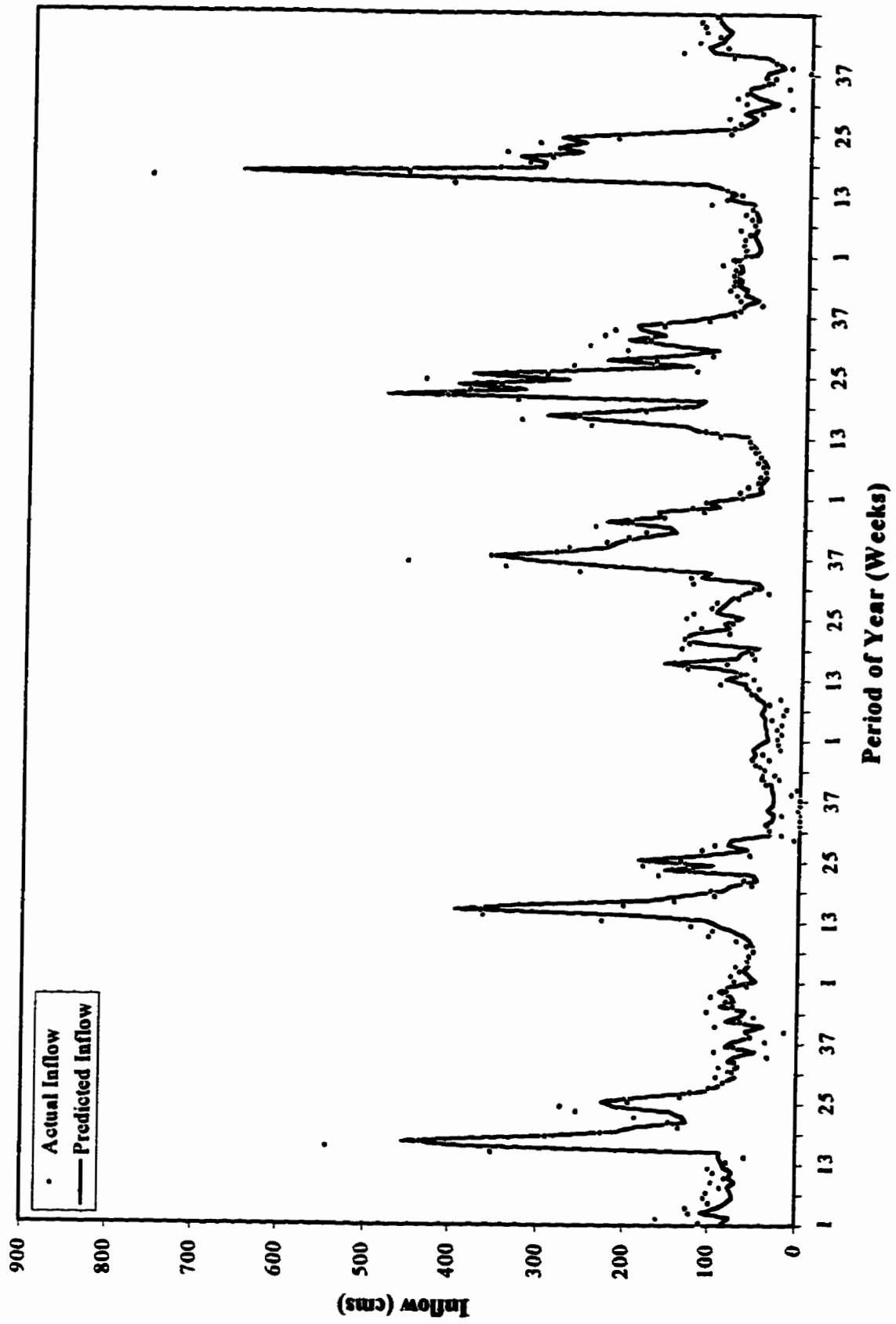


Figure F-a-3. Rainy Lake Training Results, 1975-79, one-week lead.

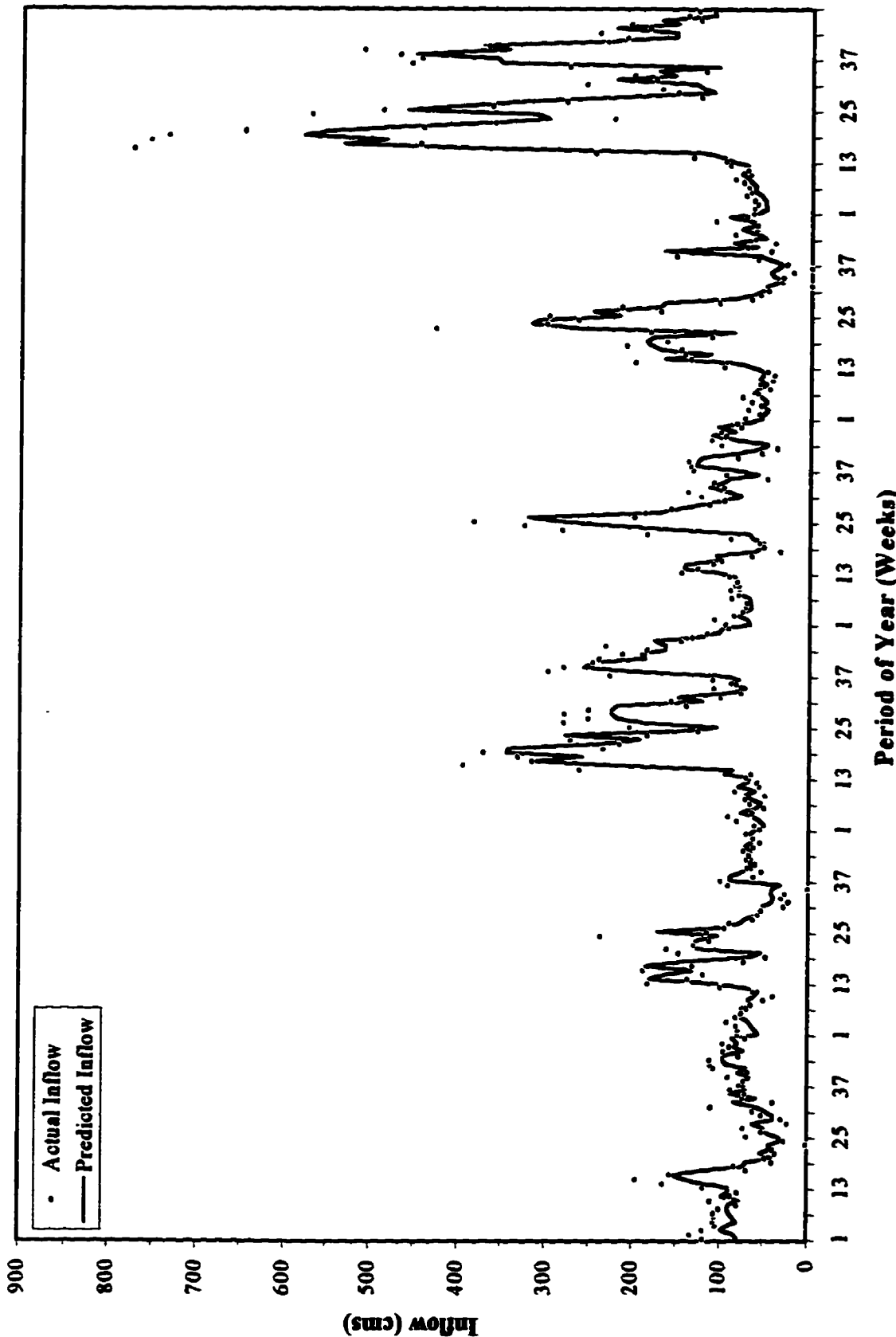


Figure F-a-4. Rainy Lake Training Results, 1980-85, one-week lead.