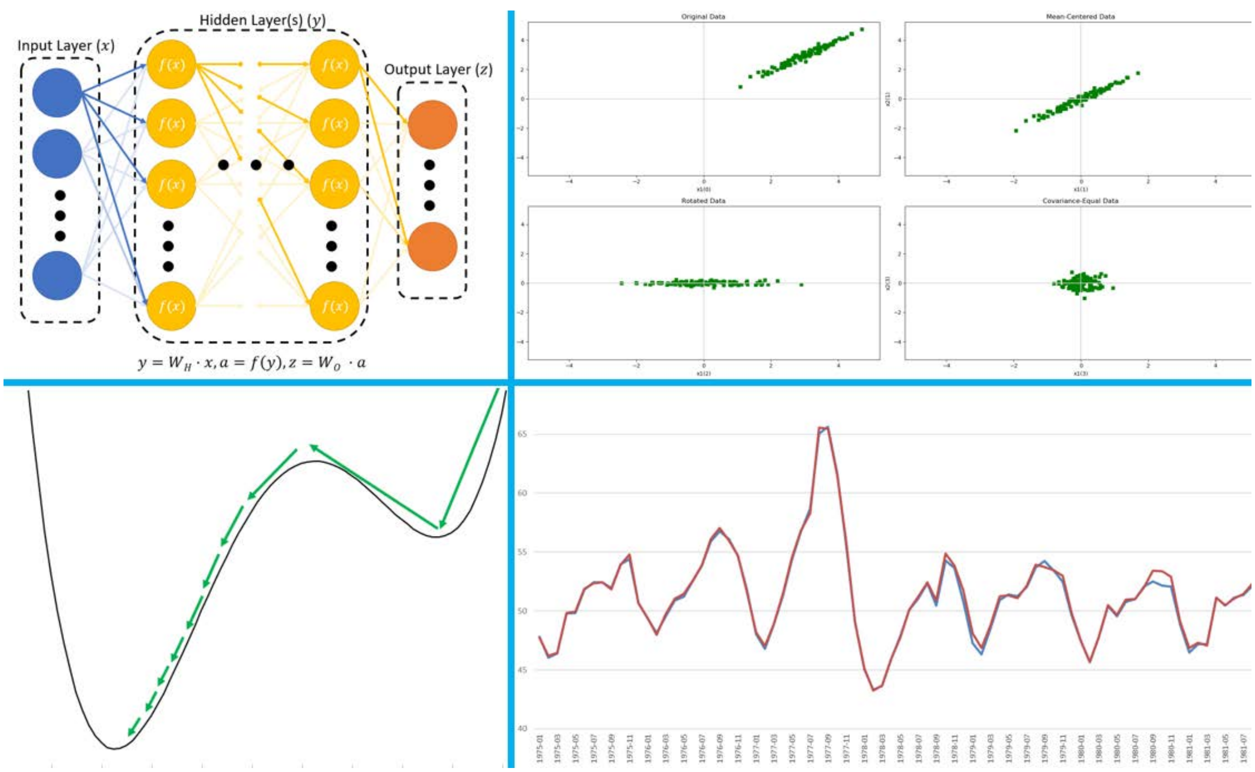
RECLAMATION

Managing Water in the West

The Implementation of Flow-Temperature Artificial Neural Network Regression into Operations Planning Models

Research and Development Office
Science and Technology Program
Final Report ST-2019-1859-01



U.S. Department of the Interior
Bureau of Reclamation
Research and Development Office

2019-09-30

Mission Statements

Protecting America's Great Outdoors and Powering Our Future

The Department of the Interior protects and manages the Nation's natural resources and cultural heritage; provides scientific and other information about those resources; and honors its trust responsibilities or special commitments to American Indians, Alaska Natives, and affiliated island communities.

REPORT DOCUMENTATION PAGE		*Form Approved* *OMB No. 0704-0188*
T1. REPORT DATE: SEPTEMBER 2019	**T2. REPORT TYPE:** RESEARCH	**T3. DATES COVERED** October 2017 – September 2019
T4. TITLE AND SUBTITLE The Implementation of Flow-Temperature Artificial Neural Network Regression into Operations Planning Models		**5a. CONTRACT NO** **5b. GRANT NUMBER** N/A **5c. PROGRAM ELEMENT NUMBER** 1541 (S&T)
6. AUTHOR(S) James Shannon, Civil Engineer/Water Resources Modeler Interior Region 10, California-Great Basin, Division of Planning 2800 Cottage Way, Room W-2830/W-2825, Sacramento, CA 95825 jshannon@usbr.gov (916) 978-5078		**5d. PROJECT NUMBER** ST-2019-1859-01 **5e. TASK NUMBER** N/A **5f. WORK UNIT NUMBER** MP-740
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) James Shannon, Civil Engineer/Hydrologic Modeler Interior Region 10, California-Great Basin, Division of Planning 2800 Cottage Way, Room W-2830/W-2825, Sacramento, CA 95825 jshannon@usbr.gov (916) 978-5078		**8. PERFORMING ORGANIZATION REPORT NUMBER** N/A
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Research and Development Office U.S. Department of the Interior, Bureau of Reclamation, PO Box 25007, Denver CO 80225-0007		**10. SPONSOR/MONITOR'S ACRONYM(S)** **R&D:** Research and Development Office **BOR/USBR:** Bureau of Reclamation DOI: Department of the Interior **11. SPONSOR/MONITOR'S REPORT NUMBER(S)** ST-2019-1859-01

12. DISTRIBUTION / AVAILABILITY STATEMENT
Final report can be downloaded from Reclamation's website: https://www.usbr.gov/research/

13. SUPPLEMENTARY NOTES

14. ABSTRACT *(Maximum 200 words)* Operation Planning Models (OPM) optimize the allocation of water in a complex system and feed information into other models, like water quality temperature models, but the results of those secondary models rarely inform OPMs to change the flow regime. This project researched Artificial Neural Networks (ANN) in order to emulate an existing water quality model and integrate in it into CalSim3 to better inform decisions on water allocation. The research yielded an ANN to quickly learn the non-linear operational relationship between flow and temperature criterion.

15. SUBJECT TERMS
Neural Network, Machine Learning, Water Quality, Operations Planning, Optimization

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT** **U**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Jim Shannon
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U			**19b. TELEPHONE NUMBER** (916) 978-5078

S Standard Form 298 (Rev. 8/98)
P Prescribed by ANSI Std. 239-18

BUREAU OF RECLAMATION

Research and Development Office
Science and Technology Program

Mid-Pacific Region, Division of Planning

Final Report ST-2019-1859-01

The Implementation of Flow-Temperature Artificial Neural Network Regression into Operations Planning Models

JAMES SHANNON

Digitally signed by JAMES SHANNON
Date: 2019.09.30 10:36:12 -07'00'

Prepared by: James Shannon, P.E.
Civil Engineer/Water Resources Modeler
Interior Region 10, California-Great Basin, Division of Planning

ZACKARY LEADY

Digitally signed by ZACKARY LEADY
Date: 2019.09.30 10:52:26 -07'00'

Peer Review: Zackary Leady
Water Resources Modeler
Interior Region 10, California-Great Basin, Division of Planning

Acknowledgements

Thank you to the following individuals for their contributions to this project:

- David Julian, formerly of Jacobs Engineering, for guidance on the update of the HEC-5Q Sacramento River Water Quality Model for CalSim3 Hydrology
- Hao Xie of the State of California Department of Water Resources, for providing an updated Water Resources Integrated Modeling System (WRIMS) that allows for use of Python functions for external calculations in CalSim3
- James Gilbert of the Bureau of Reclamation Technical Service Center, for providing a tool to streamline reading from and writing to the Army Corps of Engineers' Hydrologic Engineering Center Data Storage System (DSS) using the Python programming language.
- Zackary Leady of the Bureau of Reclamation California-Great Basin Division of Planning, for being a soundboard to this project's principal investigator.

Notices

The content of this report is only intended for research and discussion purposes. This report is not a proposal to change facility operations of the Central Valley Project nor does it guarantee an accurate representation of current operational policies of the Bureau of Reclamation regarding the Central Valley Project.

Acronyms and Abbreviations

ANN	Artificial Neural Network
BO	Biological Opinion
CFS	Cubic Feet Per Second
CVP	Central Valley Project
DWR	State of California Department of Water Resources
DSS	Army Corps of Engineers' Hydrologic Engineering Center Data Storage System
NMFS	National Marines Fisheries Service
NN	Neural Network
OPM	Operation Planning Model
RPA	Reasonable and Prudent Alternative
SCM	Source Control Management
SRWQM	Sacramento River Water Quality Model
SWP	State Water Project
TAF	Thousand Acre-Feet
TCD	Temperature Control Device
TSC	Bureau of Reclamation Technical Service Center
USBR	United States Department of the Interior, Bureau of Reclamation
WRESL	Water Resources Simulation Language
WRIMS	Water Resource Integrated Modeling System

Executive Summary

Operation Planning Models (OPMs) optimize the allocation of water in a complex system based on flow constraining criteria when determining the long-term water supply reliability of reservoir and river systems. Since OPMs are often entirely flow-based, additional models calculate non-flow qualities in the flow regime; an example is a water quality model calculating water temperature through a system. Traditionally, model communication between OPMs and water quality models move in one direction: from OPMs to water quality models. The one-way communication prevents water quality models from informing OPMs about aspects like temperature when allocating water. A "guess-and-check" approach is viable in some cases, where an engineer adjusts an OPM model running one cycle of the OPM and water quality models, but the effort quickly becomes time consuming when highly complex models take hours to execute.

This project researched the theory and practice of training and deploying Artificial Neural Networks (ANN) in order to construct an integrated surrogate model for existing uncoupled water quality models. The surrogate model would significantly cut down on model runtime with minimal cost to accuracy, allowing for water quality models to inform OPMs. Focusing on the Central Valley (CVP) and State Water Projects (SWP) in California, an ANN captured the non-linear operational relationship of water temperature given flow along the Sacramento river. Integrating the ANN into the CVP/SWP OPM, CalSim3, allows for a systematic approach to optimize the flow regime under complex temperature operation requirements.

The research yielded an updated HEC-5Q Sacramento River Water Quality Model (SRWQM) given a CalSim3 flow regime. Newly created Python scripts now provide modelers the capability to automatically perturb Shasta Reservoir releases to the Sacramento River and calculate the resulting water temperatures. With this established automated data generation process, an ANN can quickly learn the non-linear operational relationship between Shasta inflow, storage, outflow, and downstream temperature to help inform CalSim3 if the flow regime needs changing under a temperature criterion.

Following this project, the Bureau of Reclamation California-Great Basin, Division of Planning would like to expand the ANN to include additional temperature regulation criteria. Sharing this framework with CVP, SWP, and partner agency managers would assist in the constant negotiations of improving the California water system for all interested stakeholders. The framework is also able to expand into other problem areas where there is a need to establish a flow and non-flow criteria relationship.

Contents

Tables

Figures

Introduction and Background

Background

Operation Planning Models (OPMs) optimize the allocation of water in a complex system based on flow constraining criteria when determining the long-term water supply reliability of reservoir and river systems. Since OPMs are often entirely flow-based, additional models calculate non-flow qualities in the flow regime; an example is a water quality model calculating water temperature through a system. Traditionally, model communication between OPMs and water quality models move in one direction: from OPMs to water quality models. The one-way communication prevents water quality models from informing OPMs about aspects like temperature when allocating water. A “guess-and-check” approach is viable in some cases, where an engineer adjusts an OPM model running one cycle of the OPM and water quality models, but the effort quickly becomes time consuming when highly complex models take hours to execute results.

An Artificial Neural Network (ANN), emulating a water quality model and integrated into the OPM, significantly reduces model runtime in two ways. First, an ANN replicates the temperature model’s calculation of the flow regime/temperature profile relationship; the large system of linear equations and non-linear transformations represents the flow-temperature regression, which, when calibrated against a large enough dataset, computes an estimate orders of magnitude faster than a rule-based script. Second, the flow-temperature regression is set as a priority decision in the OPM, which finds the optimal flow regime for desired temperature profiles, eliminating the need to perform “guess-and-check” work. The integrated ANN provides functionality to make the OPM aware of water quality information, allowing the OPM to make more informed optimized decisions when considering non-flow criteria when allocating water.

Central Valley Project Operation Planning Models

The Central Valley (CVP) and State Water Projects (SWP) are some of the most complex water systems in the world intertied with each other. The CVP captures a major of California rain and snow runoff at Shasta Reservoir near Redding, California, and releases water down the 300-mile long Sacramento River, which outflows into the San Francisco Bay Delta before flowing into the Pacific Ocean. The SWP performs the same function with Oroville Reservoir in Oroville, California, with releases flowing down the Feather River before joining the Sacramento just north of Sacramento, California. In addition to meeting urban and agricultural demands north of the Delta, the CVP and SWP also pump water for delivery to cities and farms located south of the Delta. Figure 1 presents the infrastructure of the CVP and SWP.

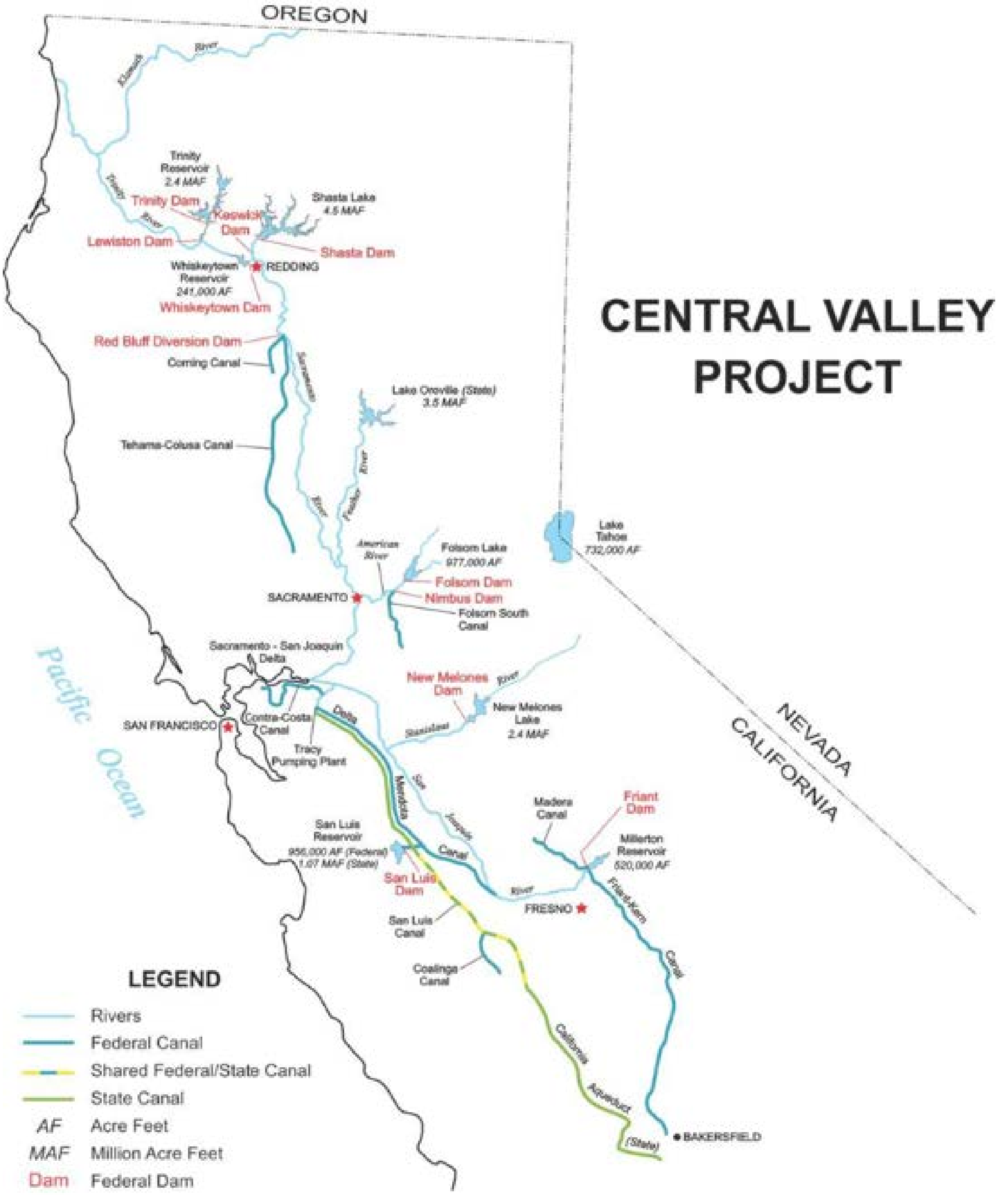


Figure 1. Map of the Central Valley Project with State Water Project infrastructure presented from usbr.gov.

In addition to meeting wholesale customer demands, the CVP and SWP must meet numerous regulating criteria imposed by partnering state and federal agencies. Biological

Opinions (BO) by the U.S. Fish and Wildlife Service (FWS) and the National Marines Fisheries Service (NMFS) impose most of these regulations with backing from the Endangered Species Act of 1973 passed by the 93rd United States Congress. These regulations include meeting minimum flow and non-flow water quality criteria at several locations throughout the system to mitigate impacts on anadromous species.

Managers and engineers utilize an OPM called CalSim3 to optimize the classically constrained problem of allocating water to demands and regulations within the CVP and SWP. CalSim3 is written in the Water Resources Simulation Language (WRESL), a domain-specific language that helps engineers to construct highly complex optimization problems for water balance systems. The over 600 WRESL text files feed into a Java-based engine, the Water Resource Integrated Modeling System (WRIMS), with time series and reference data to calculate the long-term flow regime of the CVP and SWP.

The 2009 NMFS BO on the long-term operations of the CVP and SWP established Reasonable and Prudent Alternative (RPA) I.2.1, which sets forth temperature compliance percentages for the summer season at specified locations on the Sacramento River. Table 1 presents the compliance locations and the amount of time the Sacramento River temperature shall not exceed 56ºF. CalSim3 does not have the capability to calculate temperature, so managers and engineers rely on the Sacramento River Water Quality Model (SRWQM) HEC-5Q model application to model the temperature stratification in Shasta Reservoir and simulate the operation of a Temperature Control Device (TCD) to determine how often temperature compliance is met once the flow regime is established. Because of the complexity of CalSim3 and SRWQM, these models each take about an hour each to execute results, and, since they are not integrated together, SRWQM cannot inform CalSim3 to modify the flow regime to better meet temperature compliance criteria.

Table 1: Compliance percentage for not exceeding 56ºF at select locations on the Sacramento River as specified in the NMFS BO.

Location	Compliance Percentage*
Clear Creek	95% of Time
Balls Ferry	85% of Time
Jelly’s Ferry	40% of Time
Bend Bridge	15% of Time

*Based on the 10-Year Moving Average

The State of California Department of Water Resources (DWR) previously performed a pilot study, showing that an ANN can emulate the non-linear TCD operations in SRWQM to calculate temperature in the Sacramento River, increasing runtime at minimal cost to accuracy. While the study was successful, the resulting ANN was for a much simpler flow regime and required additional outside meteorological data. This project improves upon that work, creating an ANN of similar speed and accuracy while utilizing data only available in the more complex CalSim3 flow regime.

Artificial Neural Network Theory and Practices

The idea of an Artificial Neural Network has existed since the invention of the computer in the first half of the 20th Century. An ANN attempts to replicate the functionality of neurons in the human brain, where electrical signals pass through a network of nodes and either continue or dissipate based on threshold criteria in the neural node. Mathematically, this network of nodes is represented as a system of linear equations. Weighting values are applied via a matrix dot product to input values to denote their significance, and a bias term corrects the resulting matrix via simple elementwise addition. What differentiates an ANN from a matrix is the elementwise application of an activation function to represent a threshold criterion. Figure 2 presents a conceptual schematic of an ANN's structure, and the equation below is its mathematical representation:

$$y = f(W \cdot x + B)$$

In the equation above, *x* is the input vector, with data in either row or column format, *W* are the weights applied to the input vector, *B* is the bias correction to the matrix dot product of *W* and *x*, and the function *f()* is the activation function applied elementwise to the matrix. The resulting vector is *y*, with a data format the same as the input vector *x*. All the concepts presented are in the domain of traditional linear algebra, with exception to the activation function *f()* applied elementwise. Below is a simple example to highlight an ANN procedure, with the trigonometric function sine as the activation function:

$$y = \sin(\begin{bmatrix} 0 & 0 \\ 0 & 0.5 \end{bmatrix} \cdot \begin{matrix} \pi \\ \pi \end{matrix} + \begin{matrix} \pi \\ 0. \end{matrix})$$

$$y = \sin(\begin{matrix} 0 \\ \frac{\pi}{2} \end{matrix} + \begin{matrix} \pi \\ 0. \end{matrix})$$

$$y = \sin(\begin{matrix} \pi \\ \frac{\pi}{2} \end{matrix})$$

$$y = \begin{matrix} 0 \\ 1 \end{matrix}$$

The equation provided is for a single layer ANN, but an ANN with multiple hidden layers is possible by recursively including the equation into the next layer for calculation.

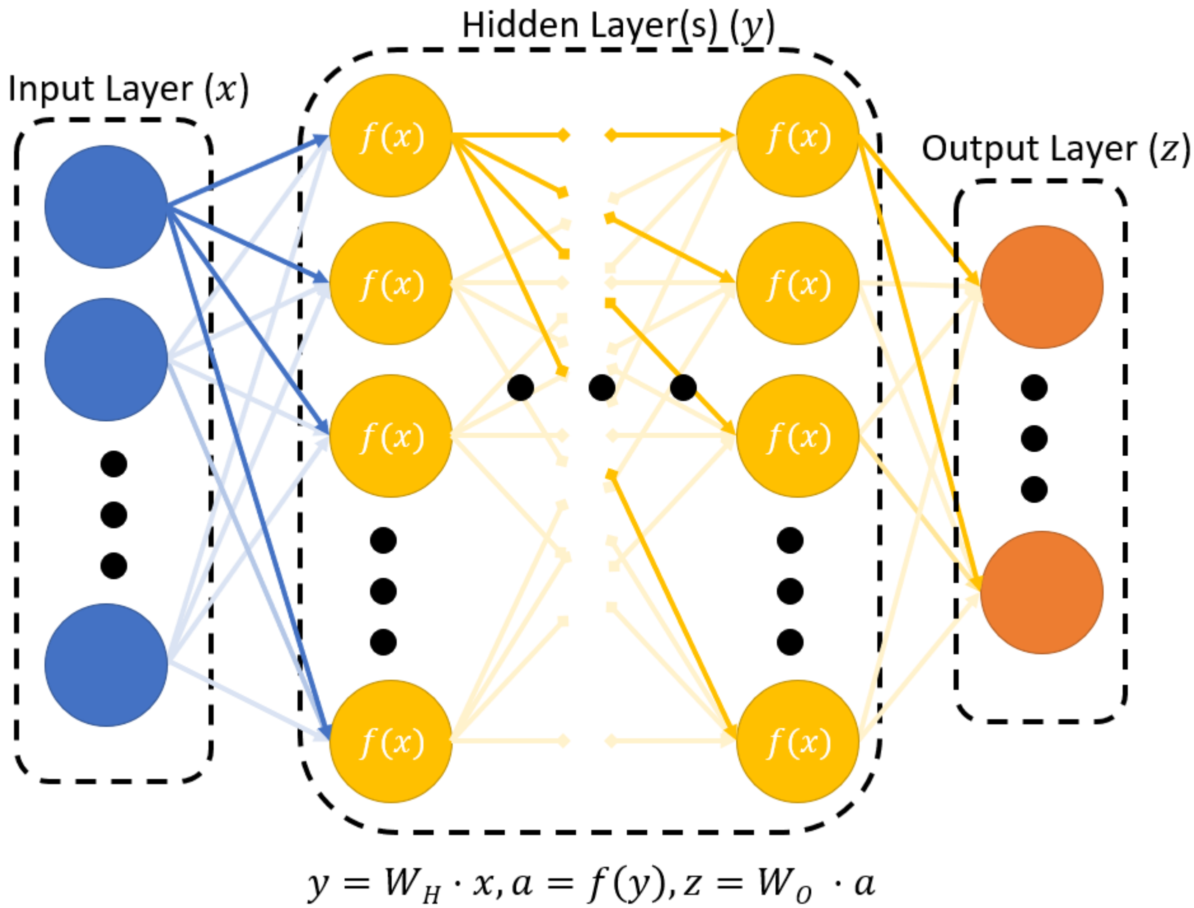


Figure 2. Conceptual Schematic of an Artificial Neural Network.

Ideally, an activation function for an ANN is a binary step threshold; the sum of all weighted inputs would either exceed a given threshold value and pass a unit value onto the next layer of weights, or the inputs would dissipate to a null value. However, a binary step activation function is a conditional statement, and, when mixed with a series of linear functions, it is difficult to solve for when ANNs have thousands of weights. Figure 3 shows two popular activation functions due to their effectiveness in improving ANN training. A hyperbolic tangent function, *tanh()*, approximates the ideal binary step function, but it can make training difficult when data varies because weight factors become diluted when values land on the asymptotic part of the curve. A rectified linear unit function, *ReLU()*, passes out the same input it receives if the sum of all weighted inputs is positive and results in a null value when a negative value is passed. The lack of an asymptotic quality allows for the most effective training, which is fined tuned with the inclusion of a bias constant. For a hyperbolic tangent to achieve a similar effectiveness, an additional training factor is required to adjust the curves amplitude, but the additional term increases complexity and thereby nullifying its effectiveness.

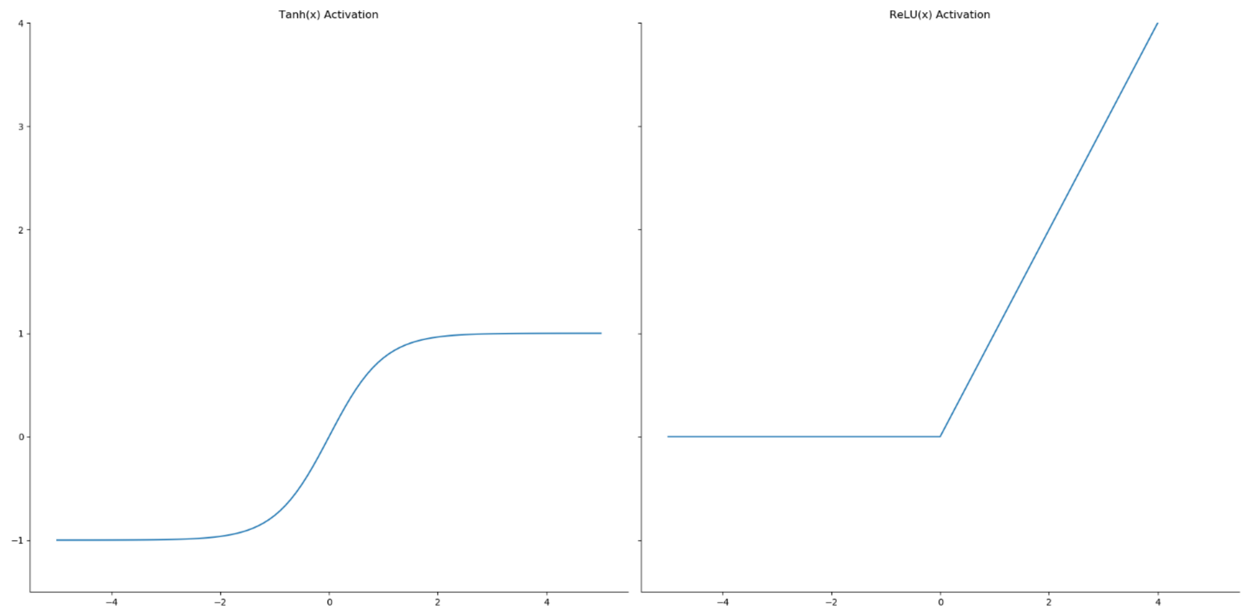


Figure 3. Plots of Hyperbolic Tangent (left) and Rectified Linear Unit Activation functions.

The goal of training an ANN is to determine the appropriate weights and biases so that it can properly emulate the model functionality. A supervised learning helps to achieve this goal. The error of a neural network is measured by the equation below:

$$E = \frac{1}{2}(y - t)^2$$

The new variables introduced are the Error *E*, and the target data *t* from our model to emulate, which is in the same data format as the ANN output *y*. The equation generations an error surface that has convex properties. Expanding the equation to include weights, biases, and inputs yields the following:

$$E = \frac{1}{2}(f(W \cdot x + B) - t)^2$$

With input data *x* and target data *t* both know values, weights *W* and biases *B* must be adjusted to minimize the error *E*, which has a lower bound of 0. However, when an ANN has hundreds to thousands of weights to adjust, the problem because too difficult. There is not enough computational power to calculate the minimum point in the error surface because of the multidimensional nature of the algorithm. Calculating the derivative of the error surface provides a method to assess in which direction the minimum point in the error surface relative to the initial conditions:

$$\frac{dE}{dWdB} = (f(W \cdot x\ +\ B) - t)f'(W \cdot x\ +\ B)(W)$$

The derivative allows an optimization algorithm to identify the weights and biases that contribute the most to the error from the target data and back propagate the relative changes needed to minimize the error.

Figure 4 illustrates this iterative process. In each training step, an optimization function calculates the derivative of the error surface to identify the steepest gradient to descend towards the minimum point in the surface. A learning constant is applied to amplify the effects of the training. If the learning constant is too large, as is the case in the red training arrows, the optimization function may miss finding the minimum point in the error surface or find it difficult to converge towards a solution. If the learning constant is too small, the optimization function may find a local minimum rather than the absolute minimum in the error surface. Some optimization functions, like the Adam Optimizer, adjust the learning rate based on information from the second derivative of the error equation and achieve effective error minimize, as represented by the green training arrows.

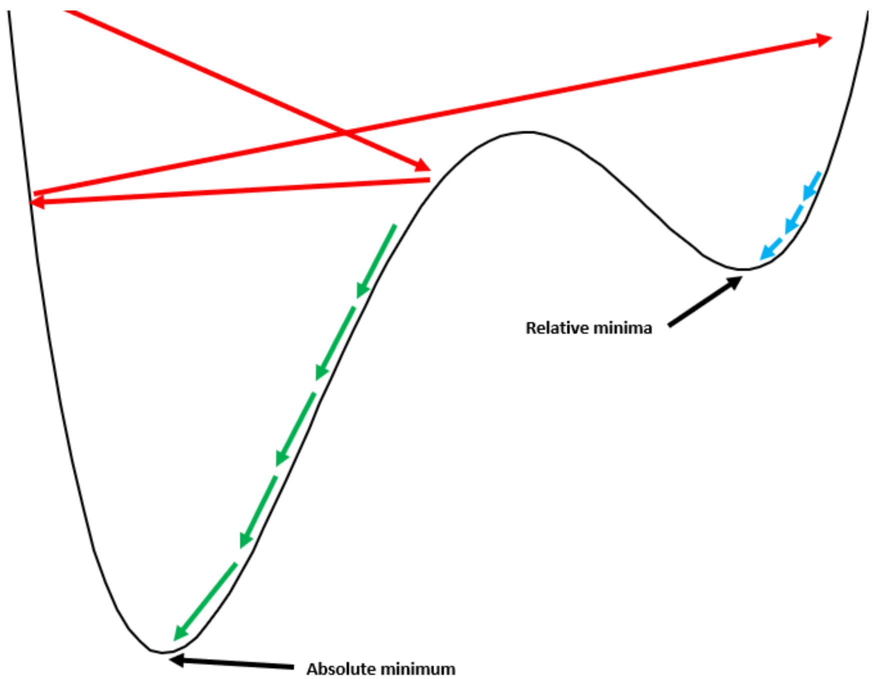


Figure 4. Conceptual rendering of ANN training to minimize error with target values.

Pre-processing the data input *x* is an additional effective method to increase training efficiency. The goal in pre-processing data is to maximum weight independence, thereby smoothing out the error surface and reducing correlation between weights during back propagation adjustments. Figure 5 illustrates the preprocessing of input data. The input data is first centered to a mean of zero to align with the activation centroid and mitigating the need for a large bias. The data is then rotated via a Principal Component Analysis, where the input values are multiplied by its eigenvalues of its covariance matrix. This step in pre-processing decorrelates the relationship between training weights. Finally, the data is rescaled to have equal variance, which transforms the error surface from an ellipsoid to a more spherical convex shape.

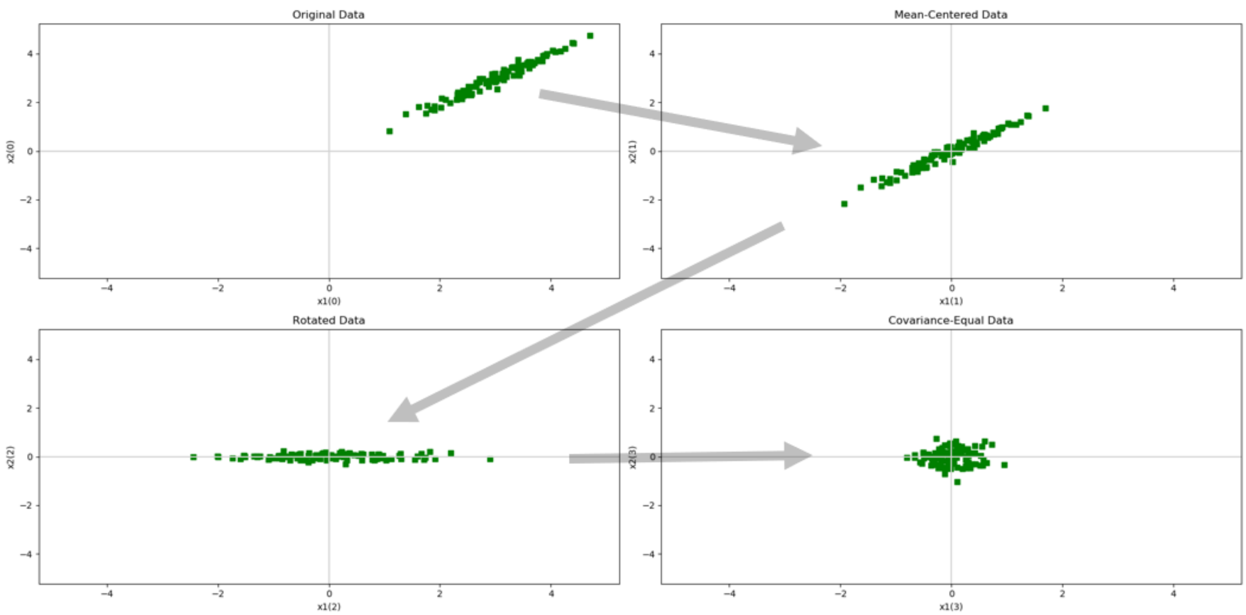


Figure 5. Preprocessing of ANN data.

As mentioned earlier, an ANN may have multiple hidden weight layers to improve its emulation of a function. Recursively including the first equation in each layer helps to achieve this attribute. Increasing the number of layers increases time and complexity for an ANN to learn the aspects of a model, but the increased number of weights allows to capture additional non-linearity. Figure 6 and Figure 7 present results of a 1-Hidden-Layer ANN and 2-Hidden-Layer ANN, respectively, in their attempts to emulate the sine trigonometric function. The 1-Hidden-Layer ANN captures the amplitude and timing of the sine function, but it has trouble replicating the output near its minimum and maximum locations where the derivative is smaller. The 2-Hidden-Layer ANN does a much better job at identifying this attribute in the curve due to its increased capacity to handle additional non-linearity.

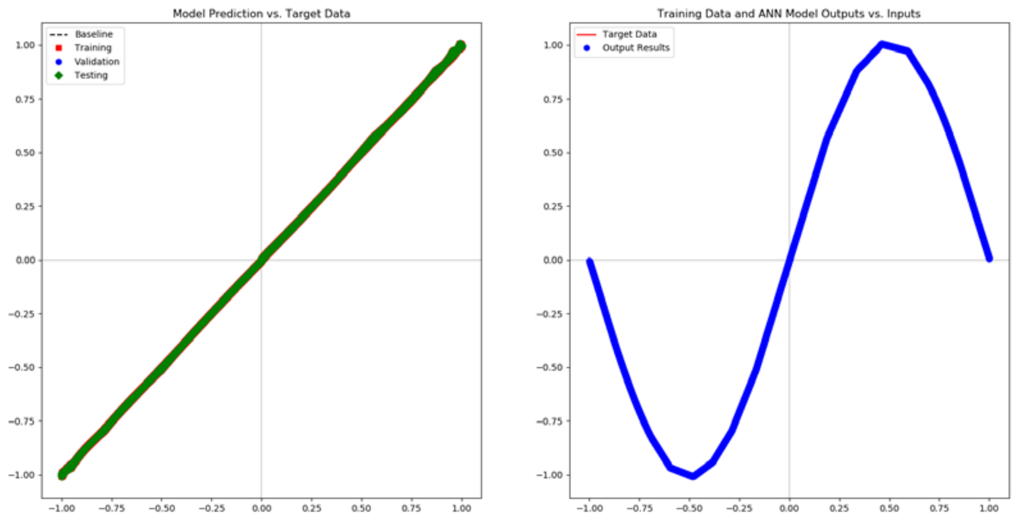


Figure 6. Results of a 1-Hidden-Layer ANN learning the function y=sin(x).

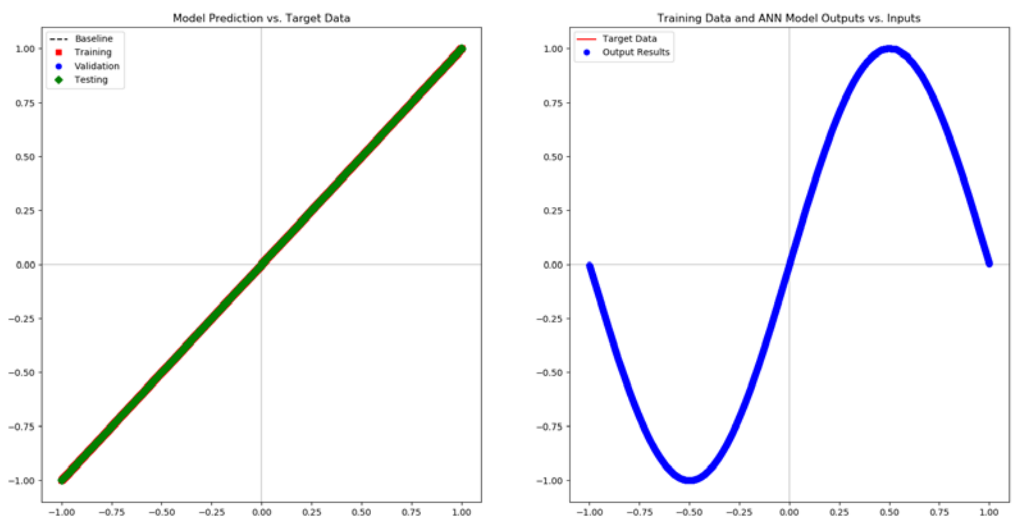


Figure 7. Results of a 2-Hidden-Layer ANN learning the function y=sin(x).

Tool Selection

This project heavily utilized the Python programming language for multiple reasons. It is a free open-source software and a highly popular tool due to its high readability and community incentivization of documentation, allowing users to create tools that are easily shareable between colleagues. The language also has built-in functionality to read and write text and data files systematically and additional capability to execute complex algorithms through its powerful “import” statement. Anaconda for Python is a third-party package management framework that allows for ease of installation and includes many popular and powerful third-party Python libraries pre-installed, such as “numpy”, to allow for quick and well-documented processing of large datasets. Additional installation of the Python library Tensorflow provided the framework to train and construct the ANN. Tensorflow offers a widely expansive library of objects and optimizers to allow for the creation of many different ANN structures. Most notably, at the time of this writing, it includes the Rectified Linear Unit activation function for ANN training, which is not available in other commercial machine learning software.

Since this project focuses on long-term temperature planning operations on the Sacramento River, CalSim3 and WRIMS were utilized to establish flow regimes, and SRWQM calculate the temperature profile targets. Prior to this project, SRWQM only accepted a CalSimII flow regime. The project chose to couple SRWQM with CalSim3 because CalSim3 offers the following functionality over CalSimII:

- CalSim3 has a greater array of wholesale customer representation than CalSimII;
- CalSim3 has a more dynamic land use representation than CalSimII’s legacy data;
- CalSim3 accounts for long-term effects on surface and groundwater interaction, whereas CalSimII lacks this functionality.

Details of the update to SRWQM to accept a CalSim3 flow regime are presented in the following sections.

Since most of the data in CalSim3, SRWQM, and Python code are in text files, the project utilized the Git Source Control Management (Git-SCM) software to track changes to text files. The software allows users to commit versions of text files and branch off versions to create sequential or parallel changes to code.

Methodology

Generating Training Data

The ANN for the project requires a CalSim3 flow regime as input and SRWQM temperature results as target data to create a successful surrogate model. For the ANN to properly inform CalSim3 of changes in temperature with respect to changes in Sacramento River flow, a variety of flow regimes were needed to capture the potential variability. A Python code utilized a functionality in WRIMS to run multiple CalSim3 studies in either parallel or sequence. The Python script, entitled ‘calsim3_runs.py’ in the data directory referenced in “Project Data Set Metadata and File Tree,” modified WRESL files to add a flow component to Shasta Reservoir releases for temperature water quality and adjusted the priority of component to perturb Sacramento River flow modeled in CalSim3. Python’s “subprocess” standard library queued model study alternatives to prevent overallocation of computer resources and called the WRIMS parallel run functionality to execute flow regime results. Figure 8 shows the additional 24 alternative flow releases from Shasta Reservoir, presenting the flow variability the ANN will learn in relation to temperature. The flow regimes typically range in a change of flow of about 1,000 CFS but can be as great as 10,000 CFS.

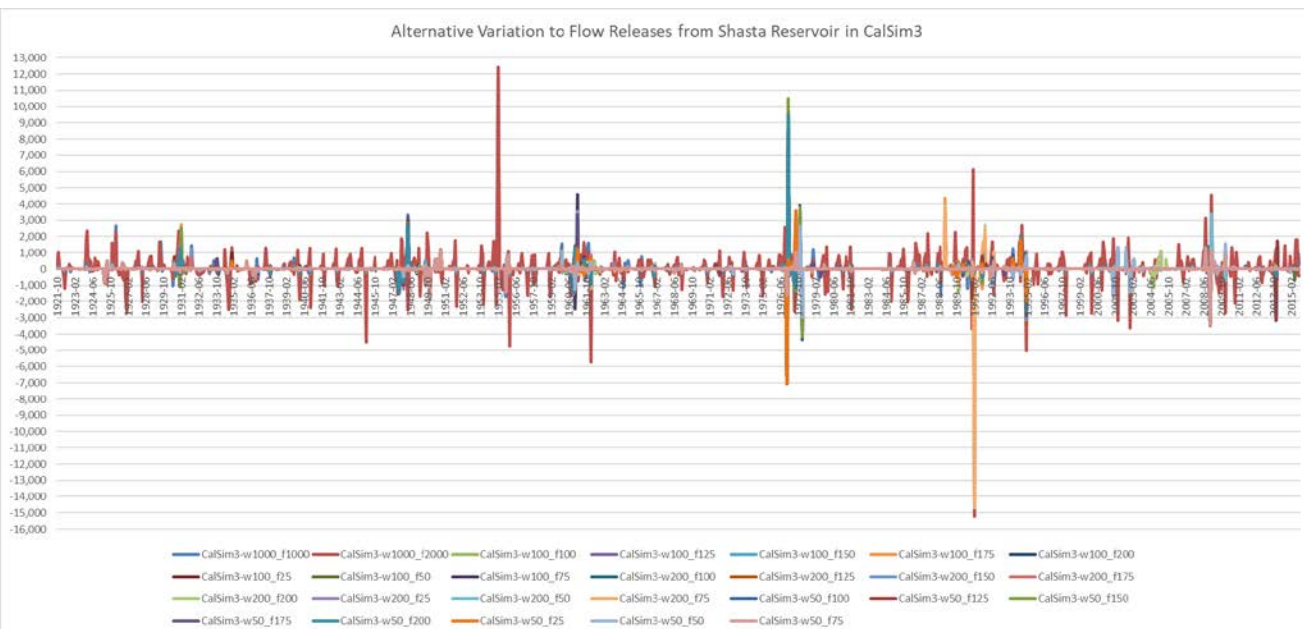


Figure 8. Plot of 24 alternative Shasta Reservoir releases represented in CalSim3 relative to a baseline.

With a variety of flow regimes established, SRWQM calculated the related temperature regime for each scenario. Prior to calculations, SRWQM had to be updated to accept a CalSim3 hydrology. The project undertook a significant effort to update the CalSimII/SRWQM mapping to CalSim3/SRWQM mapping. Lack of development documentation required iteratively changing code and reviewing results to ensure SRWQM behaved properly with the new inputs. Figure 9 and Figure 10 present excerpts of the mapping and code changes required to make the effort successful. SRWQM is now able to compute a reasonable temperature regime with

CalSim3 inputs that is like a temperature regime with CalSimII inputs but with differences that are explainable due to the differing functionality between CalSim3 and CalSimII. Results of the SRWQM model update are presented in the section "Updated SRWQM."

HEC5Q Control Point Number	HEC5Q Control Point Name	Input Types	CalSim II Node
340	Trinity Reservoir	Storage Inflow Outflow Evaporation	~~S1~~S_TRNTY ~~I1~~I_TRNTY ~~C1+F1~~C_TRNTY ~~E1~~E_TRNTY
330	Lewiston Reservoir	Inflow Diversion	~~I100~~I_LWSTN ~~D100~~D_LWSTN_CCT011
240	Whiskeytown Reservoir	Storage Inflow Outflow Evaporation	~~S3~~S_WKYTN ~~I3~~I_WKYTN ~~C3+F3~~C_CLR011 ~~E3~~E_WKYTN
220	Shasta Reservoir	Storage Inflow Outflow Evaporation	~~S4~~S_SHSTA ~~I4~~S_SHSTA ~~C4+F4~~C_SHSTA ~~E4~~E_SHSTA
200	Keswick Reservoir	Evaporation	~~E5~~E_KSWCK
180	Sacramento River below Clear Creek Confluence	Diversion	~~C5-C104~~C_KSWCK-C_SAC287-C_CLR009
178	Sacramento River below Cow Creek Confluence	Inflow	~~C10801~~C_SAC277-C_SAC289
176	Sacramento River below Cottonwood Creek Confluence	Inflow	~~C10802~~C_SAC271-C_SAC277
172	Sacramento River below Battle Creek Confluence	Inflow	~~C10803~~C_SAC269-C_SAC271
170	Sacramento River at Bend Bridge	Inflow Diversion	~~I109+R109~~C_SAC257-C_SAC269 ~~D109~~
160	Sacramento River above Red Bluff Diversion Dam	Inflow Diversion	~~C11001+I112~~C_SAC240-C_SAC257+D_SAC240_05_NA+D_SAC240_TCC001 ~~D112~~ D_SAC240_05_NA+D_SAC240_TCC001
150	Sacramento River below Woodson Bridge	Inflow Diversion	~~C11305+C11301+R113+R114A+R114B+R114C~~C_SAC217-C_SAC240+D_SAC224_04_NA ~~D113A+D113B~~D_SAC224_04_NA
140	Sacramento River at GCID	Diversion	~~D114~~D_SAC207_GCC007

Shannon, Jim... Review
Shannon, Jim... Ensure F1 typically equals 0 in
Shannon, Jim... Should this node be 300 or 320
Shannon, Jim... Ensure F3 typically equals 0 in
Shannon, Jim... Ensure D_CLR009_02_NA = 0 in
Shannon, Jim... Based on CalSim3 Schematic
Shannon, Jim... Should there be a diversion for
Shannon, Jim... Ensure F4 typically equals 0 in
Shannon, Jim... Would D_WTPJMS_03_PU1 fit
Shannon, Jim... Why is there no mapping to the
Shannon, Jim... How will we include
Shannon, Jim... Assuming this accounts for loss
Shannon, Jim... Downstream to upstream order
Shannon, Jim... Downstream to upstream order
Shannon, Jim... Downstream to upstream order
Shannon, Jim... There is no diversion in CalSim3
Shannon, Jim... Is this okay to do? Check with
Shannon, Jim... This component may need to be
Shannon, Jim... Is this okay to do? Check with
Shannon, Jim... Can I add "Inflow" to this HEC-
Shannon, Jim... This CalSim3 variable maps to

Figure 9. Schematic changes to CalSimII/SRWQM mapping to accept CalSim3 hydrology.

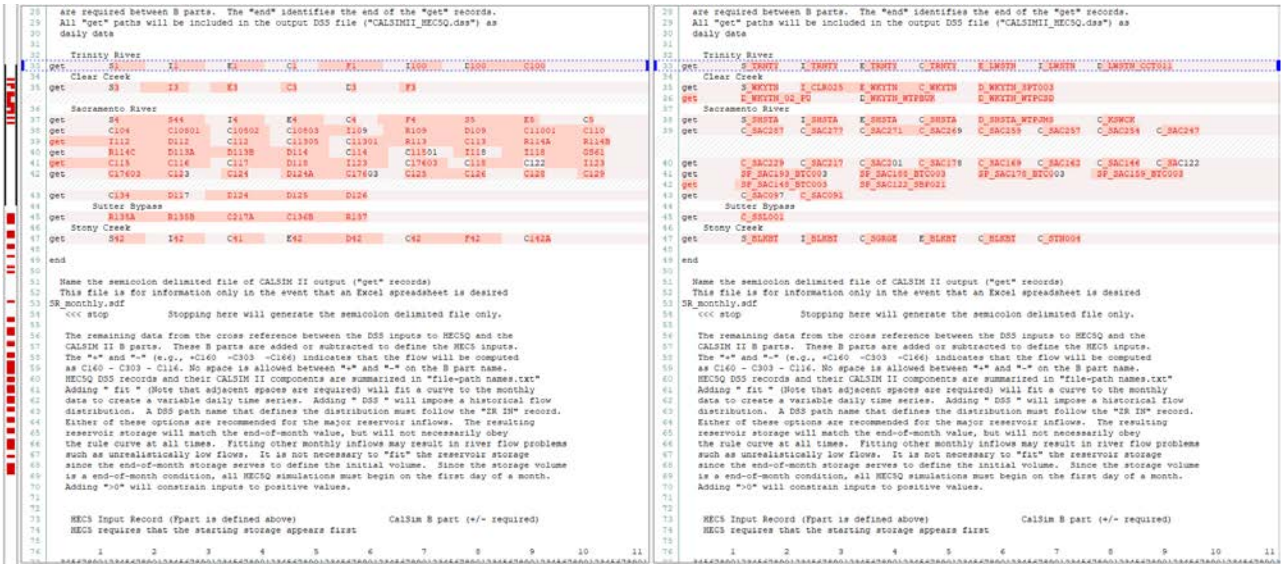


Figure 10. Code changes to SRWQM to accept CalSim3 flow regime inputs.

The multiple temperature regimes were computed in SRWQM with Python in a similar manner to how the CalSim3 flow regimes were generated. A Python script, entitled "hec5q_runs.py", queried CalSim3 flow regime and utilize executables in the HEC-5Q model toolkit to queue and run multiple studies in parallel without over-allocating computer resources.

Given flow and temperature regime scenarios, Python queried 4-month rolling window of Shasta Reservoir storage, inflow, and outflow from CalSim3 and paired the input data with Sacramento River temperature at the Clear Creek confluence. All data was aggregated on a monthly timestep for full compatibility with CalSim3. The input data was centered, rotated using Principal Component Analysis, and rescaled to equal covariance to maximize training efficiency. The input and target data were split into three groups: training, validation, and testing. The training data represented 60% of the full dataset and the other two sets each represented 20% of the full dataset. The data is then iteratively fed into a Tensorflow implemented neural network to train the weight and bias. This Python script is entitled "train_temperature_nn.py" and is listed in the data directory specified in "Project Data Set Metadata and File Tree."

Neural Network Training

A Python script utilizing the Tensorflow library allowed for the creation of the ANN training framework. The script can create an ANN of any weight size with Rectified Linear Units as its activation function. To allow for testing of different weighting structures, the script accepts a list of number of weights and hidden layers. This project looked at 4 different ANN structures:

- 1 Hidden Layer
    - Hidden Layer 1: 64 Weights
- 2 Hidden Layers
    - Hidden Layer 1: 128 Weights
    - Hidden Layer 2: 64 Weights
- 3 Hidden Layers

    - Hidden Layer 1: 256 Weights
    - Hidden Layer 2: 128 Weights
    - Hidden Layer 3: 64 Weights
- 4 Hidden Layers
    - Hidden Layer 1: 512 Weights
    - Hidden Layer 2: 256 Weights
    - Hidden Layer 3: 64 Weights
    - Hidden Layer 4: 64 Weights

The network accepted the input and target data to construct an error equation, which an Adam optimizer iteratively solved to minimize the results produced by the ANN and SRWQM given the CalSim3 inputs. The Python script entitled "FFANN.py" provides further detail of this process and is listed in the data directory specified in "Project Data Set Metadata and File Tree."

Neural Network Construction and Integration into CalSim3

DWR contributed to this project by updating WRIMS, which it maintains, to accept Python scripts as external functions to CalSim3. The update includes a new Java class object to find and connect to the Python application on a modeler's computer to interpret the provided Python script in the CalSim3 study. This functionality also requires the installation of the Jep Java module to pass Python "numpy" arrays into Java.

The Python Tensorflow script provides text files of optimized weights and biases for the ANN. A Python script utilizing the "numpy" library reads these text files to construct the ANN and is then integrated into CalSim3.

Results

Updated SRWQM with CalSim3 Hydrology

The successful update to SRWQM with the new CaSim3 flow regime resulted in similar temperature results to the previous model version. There are certainly differences with its predecessor with CalSimII flow regime as input, but those differences are due to the differing functionality between CalSim3 and CalSimII. Table 2 presents a summary of the daily disaggregation flow variables in SRWQM derived from the given monthly CalSim3 hydrology. The R^2 Score indicates a high similarity between time series magnitude and timing the closer the value is to 1, and the respective flow regime annual average values help to provide context for the R^2 Score indicator. The Shasta inflow, storage, and outflow variables used for training the ANN all have similar indicators, suggesting minimal changes when shifting regimes. Figure 11 presents the plot comparison of Shasta outflow time series to confirm this is the case. There are minor differences in peaks throughout the period of record, but the flow regimes have similar magnitude and temporal patterns, hinting that the temperature regimes should not substantially differ.

Not all variables are the same when changing flow regimes. SRWQM reservoir diversions modeling evaporation at Shasta and Keswick are lower in CalSim3 than CalSimII because CalSim3 generally has a wetter hydrology due to an updated evaporation data set. Figure 12 confirms that the temporal pattern remains the same. Diversions for agricultural demands, like “GCID Canal Diversion,” are also lower in CalSim3 due to the updated land use methodology and more robust dataset.

Table 2. Summary of flow comparison in SRWQM between CalSimII and CalSim3 flow regimes.

SRWQM Variable	R^2 Score	CalSimII Annual Average	CalSim3 Annual Average
SHASTA INFLOW	0.973365	477.94	473.202
SHASTA STORAGE	0.916514	3140.5	3094.25
SHASTA OUTFLOW	0.885965	466.881	464.54
SHASTA DIVERSION	0.773981	10.7448	8.03245
GCID CANAL DIVERSION	0.605752	59.3493	37.43
KESWICK DIVERSION	-2011	0.287475	-2.09884

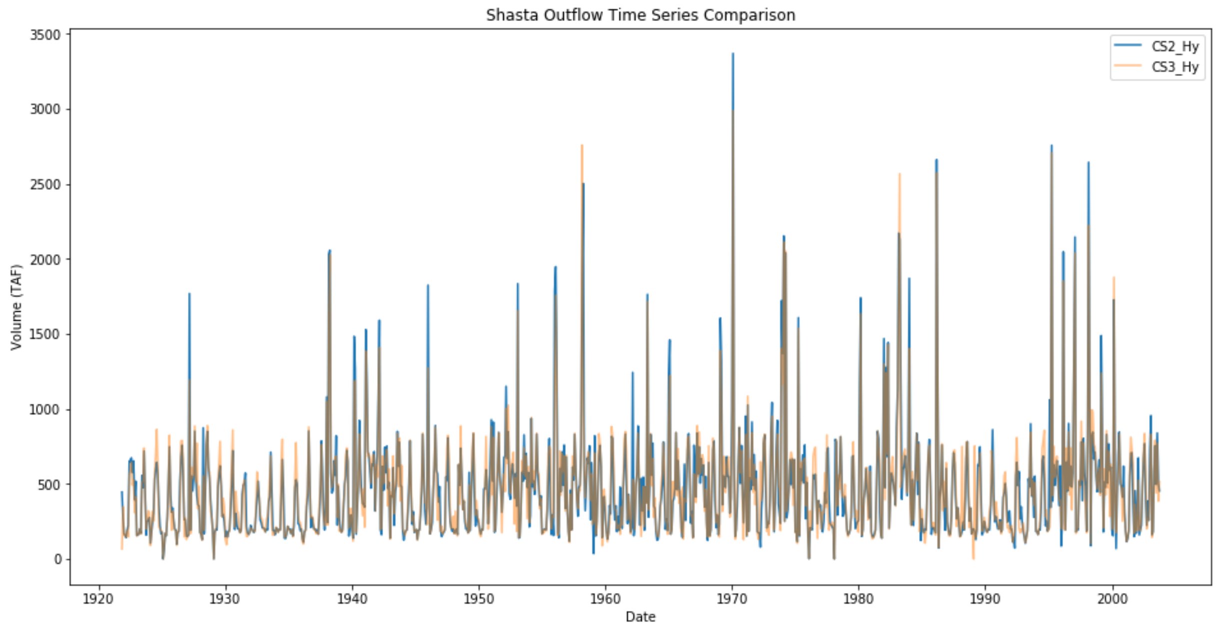


Figure 11. Daily time series comparison of Shasta Reservoir outflow releases in SRWQM given CalSimII (blue) and CalSim3 (orange) hydrologies.

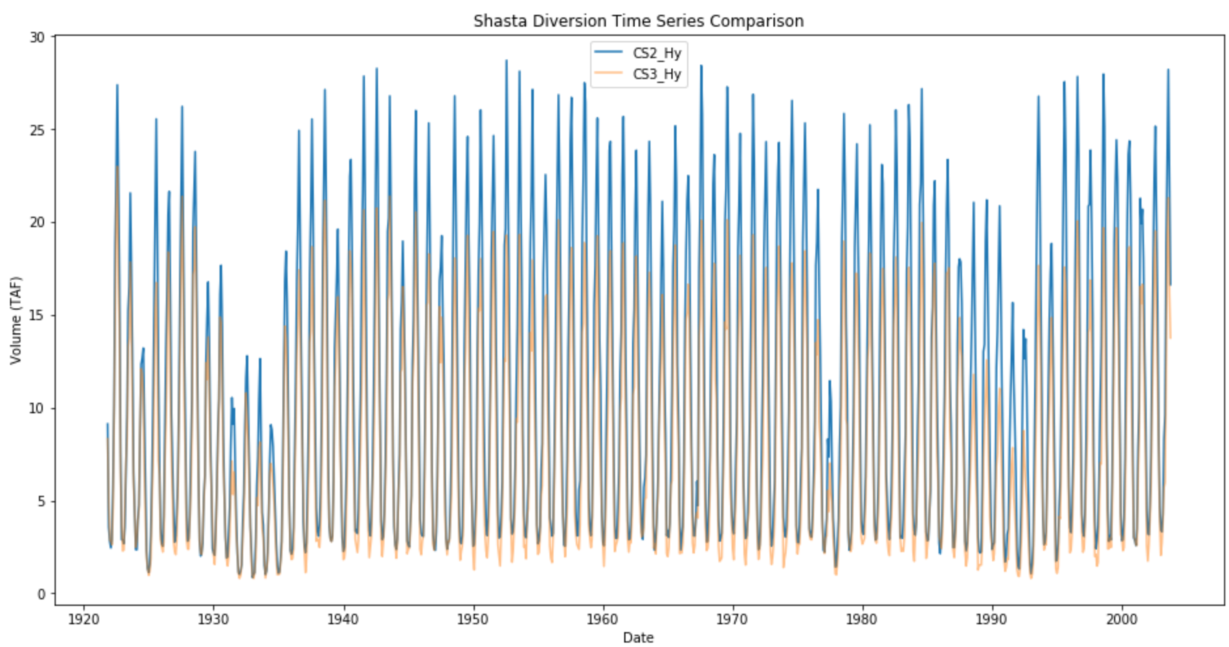


Figure 12. Daily time series comparison of direct diversions from Shasta Reservoir in SRWQM given CalSimII (blue) and CalSim3 (orange) hydrologies.

Because there were no significant differences observed between SRWQM's daily disaggregation of CalSim3 and CalSimII hydrology, Table 3 presents no significant changes in the resulting temperature regime. The upstream compliance location on Sacramento River at Clear Creek has near identical indicators between the two flow regimes, and Figure 13 confirms that the magnitude and temporal pattern are similar. The downstream compliance location on the

Sacramento River at Bend Bridge shows the same results in Table 3 and Figure 14. These similar results conclude that SRWQM is operating the TCD in Shasta Reservoir in the same manner, even with minor differences in flow regimes.

Table 3. Summary of temperature comparison in SRWQM between CalSimII and CalSim3 flow regimes.

SRWQM Variable	R^2 Score	CalSimII Annual Average	CalSim3 Annual Average
BYPASS/ABV SACRAMENTO/TEMP_F	0.996431	67.8946	67.3799
SACRAMENTO/KNIGHTS LDG/TEMP_F	0.986776	60.6017	60.7182
SACRAMENTO/BUTTE_CITY/TEMP_F	0.979937	58.1057	58.0989
CLEAR CREEK/ABV SACRAMENTO/TEMP_F	0.970932	51.8473	51.9842
SACRAMENTO/WOODSON BRIDGE/TEMP_F	0.969656	55.462	55.4163
SACRAMENTO/RED BLUFF DAM/TEMP_F	0.963912	54.436	54.4509
SACRAMENTO/BEND BRIDGE/TEMP_F	0.955015	53.5691	53.6332
SACRAMENTO/JELLYS FERRY/TEMP_F	0.941926	53.0533	53.2545
SACRAMENTO/BALLS FERRY/TEMP_F	0.929858	52.2682	52.4448
SACRAMENTO/BLW CLEAR CREEK/TEMP_F	0.924276	51.5493	51.7686
SACRAMENTO/BLW KESWICK/TEMP_F	0.916939	50.6276	50.9005
SACRAMENTO/BLW SHASTA/TEMP_F	0.898795	49.2794	49.7305

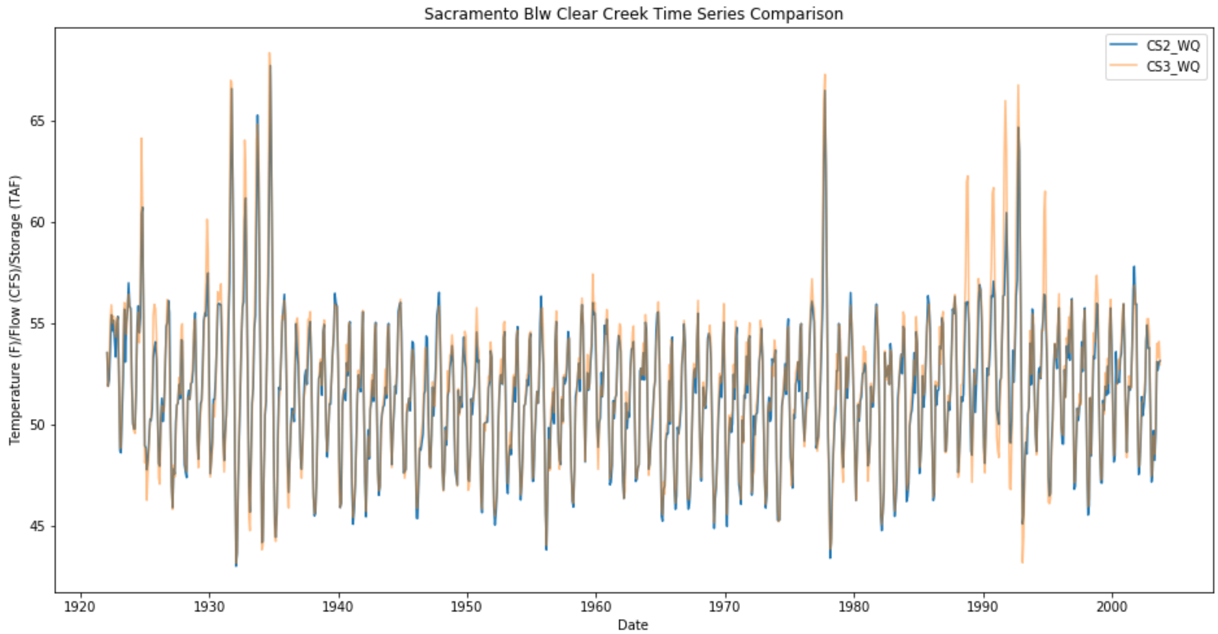


Figure 13. Daily time series comparison of Sacramento River temperature at the confluence with Clear Creek in SRWQM given CalSimII (blue) and CalSim3 (orange) hydrologies.

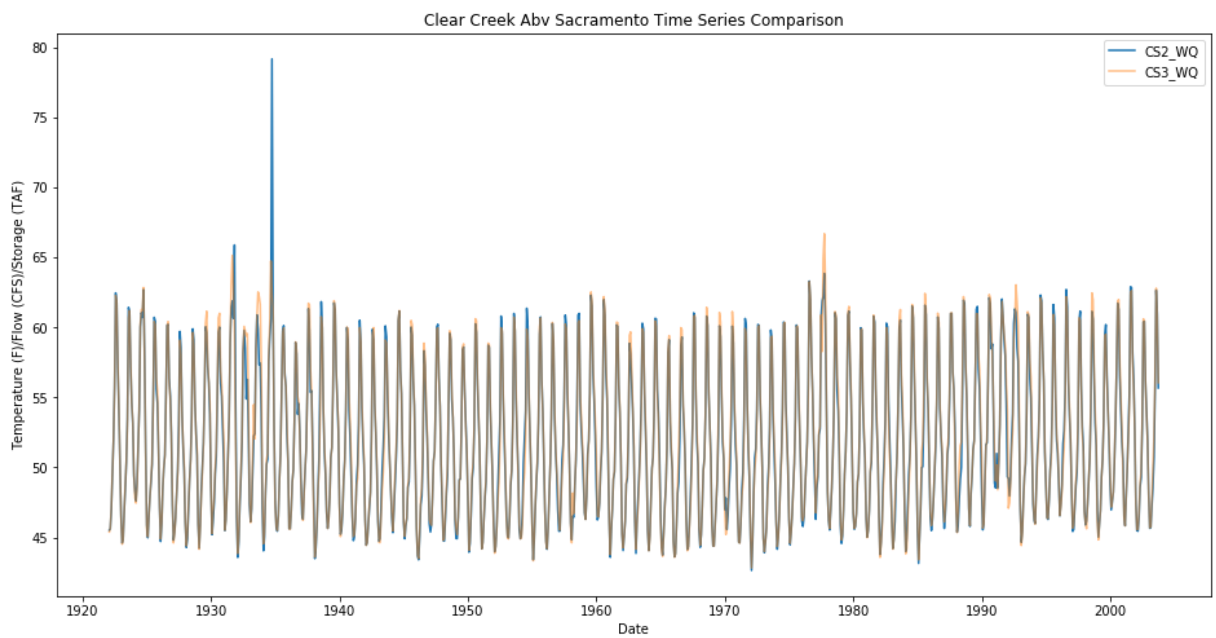


Figure 14. Daily time series comparison of Sacramento River temperature at Bend Bridge in SRWQM given CalSimII (blue) and CalSim3 (orange) hydrologies.

Neural Network Results Compared to HEC-5Q Results

Four different ANNs trained against the CalSim3 flow regime and SRWQM temperature target data. Each ANN had the same activation data, activation function, and optimizer and only varied in number of hidden layers and weights. Table 4 presents the accuracy of each ANN, where a value of 1 represents perfect accuracy. As the number of training weights expanded from

1 hidden layer to 4 hidden layers, the accuracy of the ANN results compared to SRWQM results significantly increased. Table 4 also presents the ANN accuracies with the validation and testing data set; the similar results with the training data set shows that the ANN did not overfit to the data provided and is able to calculate a result with values within its input domain.

Table 4. Comparison of daily flow results in SRWQM between CalSimII and CalSim3 flow regimes.

Artificial Neural Network Structure	Accuracy on Training Data	Accuracy on Validation Data	Accuracy on Testing Data
1 Hidden Layer Hidden Layer 1: 64 Weights	0.9179	0.9162	0.9106
2 Hidden Layers Hidden Layer 1: 128 Weights Hidden Layer 2: 64 Weights	0.9749	0.9728	0.9643
3 Hidden Layers Hidden Layer 1: 256 Weights Hidden Layer 2: 128 Weights Hidden Layer 3: 64 Weights	0.9884	0.9859	0.9832
4 Hidden Layers Hidden Layer 1: 512 Weights Hidden Layer 2: 256 Weights Hidden Layer 3: 64 Weights Hidden Layer 4: 64 Weights	0.9911	0.9893	0.9883

Figure 15 respectively present comparison scatter and time series plots between the ANN with 1 Hidden Layer and SRWQM. The ANN decently models the temporal and magnitude patterns in the SRWQM data but is unable to calculate temperature within 1ºF because it lacks sufficient non-linearity. Figure 16 shows the same type of plots comparing the ANN with 4 Hidden Layers and SRWQM results, highlighting the much-improved accuracy of estimating temperature within 0.25ºF because of the increased non-linearity from additional activation functions.

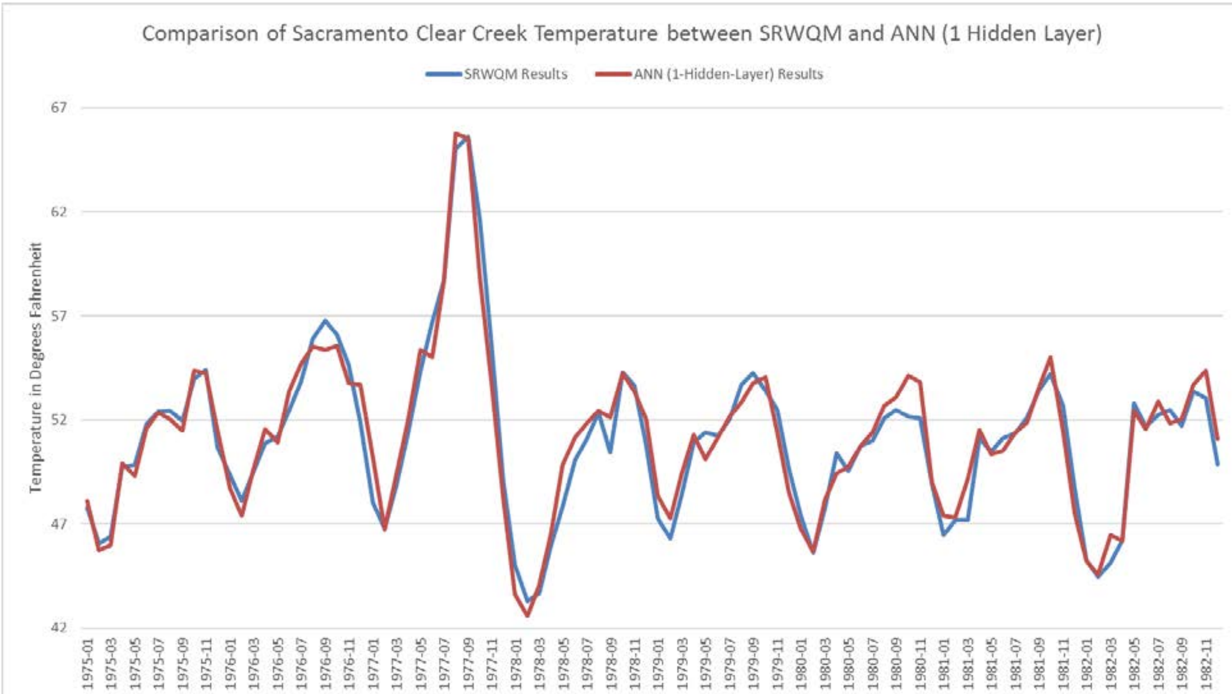


Figure 15. Time series plot of ANN (1 Hidden Layer) results compared to target data from SRWQM.

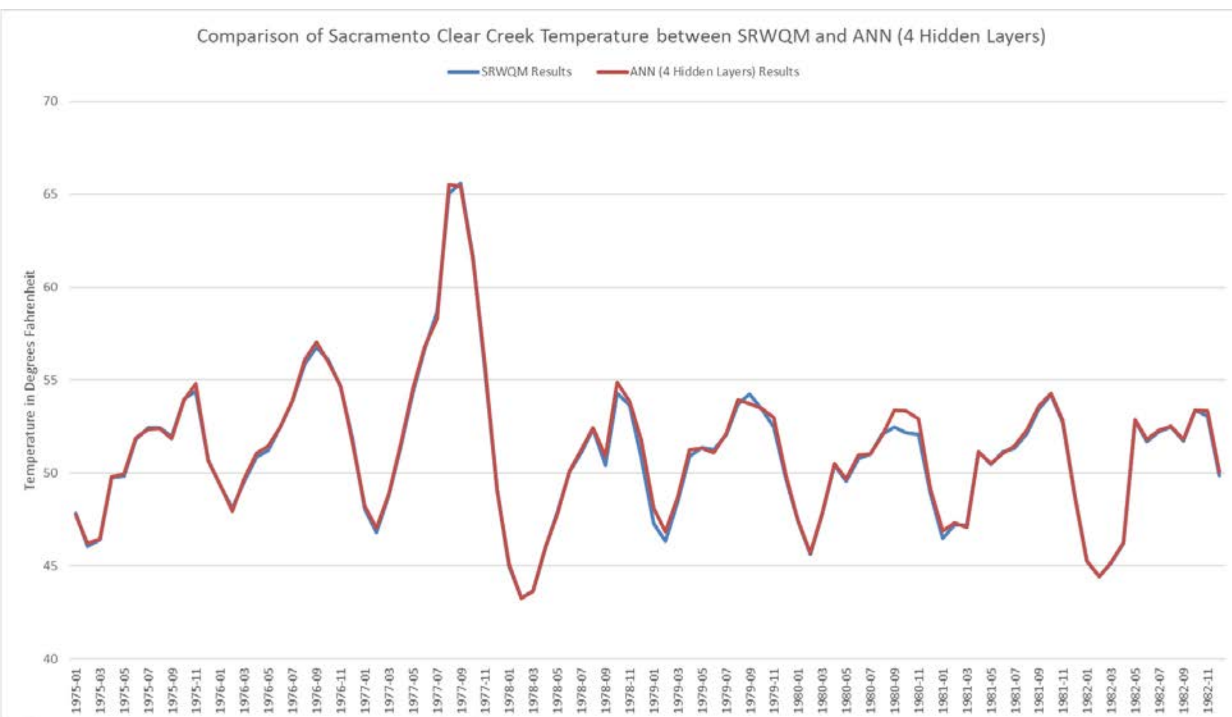


Figure 16. Time series plot of ANN (4 Hidden Layer) results compared to target data from SRWQM.

Recommendations

Future Improvements

The newly constructed ANN integrated into CalSim3 allows managers to negotiate changes in the regulatory environment over the CVP and SWP with knowledge regarding temperature water quality. Requests are already made to expand the ANN to include the other temperature compliance locations downstream of Clear Creek and update CalSim3 model operations to greater utilize the ANN.

The Python script utilized Tensorflow version 1, and Tensorflow version 2 is recently released at the time of this writing. Updating to version 2 would lead to reduced training times based on newly available features and increase code readability and maintainability. During this update, the Python code will be updated to handle any generic dataset rather than focus primarily on CalSim3 and SRWQM data. Some discussion has begun of utilizing this framework to understand DWR’s development of the Delta Salinity ANN that exists in CalSim3 and CalSimII to improve quality control review.

The training and integration of an ANN is certainly applicable into other OPMs. Python allows for the direct embedding into C/C++, R, Julia, and Fortran applications, and its “subprocess” standard library allows for connecting memory pipelines between models, if full integration is not possible.

Conclusions

This project successfully constructed an Artificial Neural Network (ANN) to simulate the non-linear relationship between the Central Valley and State Water Projects' CalSim3 flow operation regime and the Sacramento River Water Quality Model's temperature operation machine. The trained ANN integrated into CalSim3 allows the long-term operational planning model to make flow decisions with knowledge of the temperature in the Sacramento River within 0.25ºF given a release from Shasta Reservoir. The framework presented in this report heavily utilize the Python programming language to well-document the process of training and constructing an ANN so that it can replicate other Operation Planning Models and related secondary models.

Bibliography

Chacon, Scott, and Ben Straub. n.d. *Git - Book.* Accessed January 16, 2019. https://git-scm.com/book/en/v2.

Hsu, En-Ching, and Prabhjot Sandhu. 2013. *Temperature Model Development for CalSim.* Sacramento, June.

Karpathy, Andrej. n.d. *CS231n Convolutional Neural Networks for Visual Recognition.* Accessed February 8, 2018. http://cs231n.github.io/neural-networks-2/.

Lawrence, Steve, C. Lee Giles, and Ah Chung Tsoi. 1996. *What Size Neural Network Gives Optimal Generalization?* Princeton, August.

LeCun, Yann, Leon Botton, Benevieve B. Orr, and Klaus-Robert Muller. 1998. *Efficient BackProp.* Berlin.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. *Deep learning.* New York, May 28.

Loucks, Daniel P., and Eelco Van Beek. 2005. *Water Resources Systems Planning and Management: An Introduction to Methods, Models, and Applications.* Paris.

McCaffrey, James. 2015. *Neural Network Train-Validate-Test Stopping.* May 13. Accessed December 22, 2017. https://visualstudiomagazine.com/articles/2015/05/01/train-validate-test-stopping.aspx.

National Marine Fisheries Service Southwest Region. 2009. *Biological Opinion and Conference Opinion on the Long-Term Operations of the Central Valley Project and State Water Project.* Long Beach, June 4.

Nielsen, Michael. n.d. *Improving the way neural networks learn.* Accessed December 22, 2017. http://neuralnetworksanddeeplearning.com/chap3.html.

Python Software Foundation. 2018. *Python 3.7.0 Documentation.* June 27. Accessed September 28, 2018. https://docs.python.org/3/.

Rolnick, David, and Max Tegmark. 2018. *The Power of Deeper Networks for Expressing Natural Functions.* Boston, April 27. https://arxiv.org/pdf/1705.05502.pdf.

Ruder, Sebastian. 2017. *An overview of gradient descent optimization algorithms.* Dublin, June 15.

State of California Department of Water Resources; U.S. Bureau of Reclamation. 2017. *CalSim 3.0 Draft Report: A Water Resources System Planning Model For State Water Project (SWP) & Central Valley Project (CVP).* December.

Tch, Andrew. 2017. *The mostly complete chart of Neural Networks, explained.* August 4. Accessed July 30, 2018. https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464.

TensorFlow. n.d. *TensorFlow Core.* Accessed April 12, 2017. https://www.tensorflow.org/overview/.

The MathWorks, Inc. n.d. *Neural Network Toolbox.* Accessed February 24, 2017. http://matlab.izmiran.ru/help/toolbox/nnet/backpr59.html.

U.S. Army Corps of Engineers Hydrologic Engineering Center. 1998. *HEC-5 Simulation of Flood Control and Conservation Systems.* Davis, October.

Appendix A – Project Data Set Metadata and File Tree

The following is meta-data describes the files and data associated with this project.

- Data is stored in the following ZIP file on MP-740's share drive: J:\Staff_folders\jshannon\Backup\usbr_temperature_ann.zip
- Point of Contact:
  - Name: James Shannon
  - Email: jshannon@usbr.gov
  - Direct Phone #: 916-978-5078
  - Division Phone #: 916-978-5060
- Predominant file types:
  - Python script (*.py)
  - U.S. Army Corps of Engineers' Hydrologic Engineering Center Data Storage System (*.dss)
  - WRESL files (*.wresl)
  - WRESL table files (*.table)
- Keywords: CalSim3, HEC5Q, ANN, toolkit
- Approximate total size of all files: 90.5 GB (27.2 GB after compression)

Below are the top-level directories and descriptions, which include all files and data associated with this project in the aforementioned ZIP file:

- **calsim_toolkit** (194 MB): Updated version of query tool for reading and writing CalSim3 and SRWQM data with Python.
- **CalSim3_Studies** (13.3 GB): CalSim3 studies containing flow variation of Shasta reservoir releases for ANN training.
- **cs_otfa** (5.05 MB): Legacy version of query tool for reading and writing CalSim3 and SRWQM data with Python.
- **HEC5Q** (76.9 GB): Toolkit containing SRWQM studies of resulting temperature based on input from "CalSim3_Studies" for ANN training.
- **usbr_temperature_ann** (239 MB): Python and Tensorflow scripts used to train and construct the temperature ANN.