Jun 26th, 2:00 PM - 3:20 PM

Framework-enabled Meta-Modeling

Francesco Serafin
*University of Trento*, francesco.serafin.3@gmail.com

Olaf David
*Colorado State University - Fort Collins*, odavid@colostate.edu

Andre Dozier
*Colorado State University*

Jack Carlson
*Colorado State University - Fort Collins*

Follow this and additional works at: [https://scholarsarchive.byu.edu/iemssconference](https://scholarsarchive.byu.edu/iemssconference)


This Oral Presentation (in session) is brought to you for free and open access by the Civil and Environmental Engineering at BYU ScholarsArchive. It has been accepted for inclusion in International Congress on Environmental Modelling and Software by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.
Abstract: Applications of physically-based environmental models originating from research should be ubiquitous to use in both research and planning/consulting environments. However, due to their complexity, data resolution requirements, parameter number, platform affinity, and other criteria they are rarely suited “out-of the box” for field and consulting applications. Results from physically-based models are considered most accurate but operating an entire system requires dedicated knowledge, extensive set up, and sometimes significant computational time. Questions from field applications conversely require easy to get, quick and “accurate enough” answers. As a result, attempts to use physically-based research model in the field has caused problems for timely delivery of research models, IT deployment infrastructure management, model usability for the field user, performance expectations, data provisioning for field use, and field user training. The use of web-services might alleviate some of the implications for model users but ultimately shift the responsibility and workload to the hosting environment. This contribution proposes a machine learning (ML)-based meta model approach aiming to capture the intrinsic knowledge of a physical model into an ensemble system of artificial neural networks and make it available for providing simplified answers to on the field problem-specific questions. A meta modeling approach was developed to help transitioning from research to field by enabling a modeling framework to interact with ML libraries to emerge model surrogates a(ny) modelling solution. The Cloud Services Integration Platform CSIP/OMS was extended and utilized to harvest data and derive the meta-model at the modeling framework level. Here, NeuroEvolution of Augmenting Topology (NEAT) techniques in an ensemble application, combined with ANN uncertainty quantification and other approaches are the main methodologies used. Two examples applications have been prototyped and will be presented, a sheet and rill erosion model and a daily runoff model.

Keywords: machine learning; NEAT; meta-model; field application.

1 INTRODUCTION

Physically based models, output of research environment efforts, became over time complex softwares that allow for accurately answering scientific questions. However, daily use and on field applications of these tools are made difficult by a number of issues. First of all, the setup of the entire IT infrastructure requires expensive dedicated hardware and high level expertise personnele focused on software maintenance and deployment.

Secondly, before properly exercising a physical model scientists must go through the steps of collecting sufficient high resolution data, convert them into proper format and set up a proper calibration process. Especially the last step is a complex task that requires in-depth model knowledge and high level understanding of implemented equations. It is also computational expensive and time consuming.

Once the entire system is set up a simple simulation run is sometimes very computationally expensive and time consuming as well.

On the field personnele is not trained to deal with all the complexities just described. They mostly need quick and “accurate enough” answers providing limited input parameters.
This abstract proposes a framework enabled meta-modelling approach to harvest and encapsulate specific aspects of a physical model into an ensemble of artificial neural networks. The trapped knowledge is then exercised requiring few input information and providing uncertainty quantified results.

2 METHODOLOGICAL APPROACH

There is a gap between mandatory requirements to exercise models developed in research environments and information actually available in planning environments. Most of the time on the field personnel can’t meet all the requirements and are then not able to get needed answers. To bridge this gap, there is the need of a new tool able to provide answer with limited information. A new layer between physical models and planning environment is necessary. Because a model is a simplified representation and abstraction of a physical system, the new layer is a meta-model that is an even higher level of model abstraction. It is a model of a model, able to learn and reproduce the behaviour of a specific process of a physical model.

This abstract proposes NeuroEvolution of Augmenting Topologies (NEAT) (Stanley et al. (2002)) as learning algorithm to train artificial neural networks to mimic physical model behaviour. The end product is a lightweight black-box tool able to answer specific questions with limited input data.

3 TECHNICAL APPROACH AND IMPLEMENTATION

The framework is built around FS-NEAT (Whiteson et al. (2005)) capabilities of adapting weights and structure of the neural network during the training phase. It is also developed and deployed upon OMS-CSIP modeling framework and made available as a sequence of five CSIP services. The first service collects raw data into a dedicated MongoDB collection for later ann training. Data preparation is obviously application specific and requires expertise and in-depth model knowledge in order to select only the most sensible and useful input parameters.

The second service simply hits MongoDB database with and aggregate command in order to run the normalization algorithm database-side. This creates a further collection of normalized data and relevant statistics for later denormalization process. This implementation move the computational burden database-side avoiding the slower processes of copying data to the running service, normalizing data and then transferring normalized data back to the database.

The third service drives the training phase: retrieves data from normalized collection, splits the dataset into training and validation, trains the neural network, validates it and stores network and related validation statistics into the database. Because of the genetic algorithm, every time the neural network is trained with a different weights matrix and structure. This is the reason why, hitting this service several times allows for generating a bunch of different neural networks, each with its own capability of emulating the physical model behaviour.

The fourth service goes through the statistics of every trained and validated neural network, picks the most performant ones and stores their IDs into another collection.
The actual exposed to the on field personnel service is the fifth one: it just requires limited information from the user, retrieves the ensemble of neural networks, feeds the input node of each neural network, runs all them and returns denormalized uncertainty quantified results.

REFERENCES
