Physics-aware Architecture of Neural Networks for Uncertainty Quantification: Application in Lake Temperature Modeling

Arka Daw
darka@vt.edu
Virginia Polytechnic and State University
Blacksburg, Virginia

Anuj Karpatne
karpatne@vt.edu
Virginia Polytechnic and State University
Blacksburg, Virginia

ABSTRACT

In this paper, we explore a novel direction of research in theory-guided data science to develop *physics-aware architectures* of artificial neural networks (ANNs), where scientific knowledge is baked in the construction of ANN models. While previous efforts in theory-guided data science have used physics-based loss functions to guide the learning of neural network models to generalizable and physically consistent solutions, they do no guarantee that the model predictions will be physically consistent on unseen test instances, especially after slightly perturbing the trained model, as explored in dropout using testing methods for uncertainty quantification (UQ). On the other hand, our physics-aware ANN architecture hard-wires physical relationships in the ANN design, thus resulting in model predictions which always comply with known physics, even after performing dropout during testing for UQ. We provide some initial results to illustrate the effectiveness of our physics-aware neural network architecture in the context of lake temperature modeling, and show that our approach shows significantly lower physical inconsistency as compared to baseline methods.

KEYWORDS

physics aware architecture, theory-guided data science, neural networks, uncertainty quantification, lake temperature modelling, Long Short Term Memory, LSTM

ACM Reference Format:
Arka Daw and Anuj Karpatne. 2021. Physics-aware Architecture of Neural Networks for Uncertainty Quantification: Application in Lake Temperature Modeling. In *Proceedings of ACM Conference (Feed Workshop SIGKDD'19).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

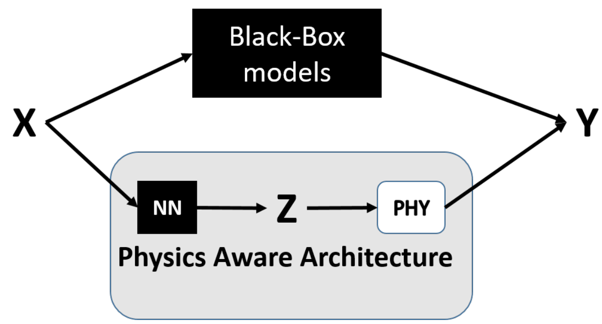


Figure 1: A schematic illustration of Physics-aware architecture of neural networks, where a first block of hidden layers (NN) maps the input X to intermediate physical variables Z, which can then be mapped to the target variable Y using physics (PHY).

1 INTRODUCTION

In the last decade we have witnessed a massive explosion of data, and data science models (e.g., deep learning methods) have been used in a myriad of applications such as remote sensing, fraud detection, financial banking, healthcare, and bio-informatics. In the light of these advancements, there is a growing interest in the scientific community to unlock the power of data science methods in scientific and engineering applications, where there is a rich background of scientific knowledge available in the form of domain theories. Despite this interest, "black-box" data science models, which are developed and deployed in a manner agnostic to underlying scientific theories (and are *de facto* standards in commercial applications of data science), face several challenges when applied in scientific problems. This is primarily due to their sole dependence on representative labeled data for training which is often difficult to obtain in real-world scientific applications. In particular, a black-box data science model for a supervised learning problem can only be as good as the representative quality of the labeled data it is fed with. To address these challenges, there is a growing body of work in the emerging field of "theory guided data science," (TDGS) [5] that aims to combine scientific knowledge with data science methods to produce generalizable as well as physically consistent solutions. For example, a recent line of work [4, 6] explored the use of physics-based loss functions to guide the learning of artificial neural network (ANN) models to be consistent with known physical relationships between the target variable and other physical variables.

While the introduction of physics-based loss functions have been shown to improve the generalizability as well as scientific consistency of ANN models, the choice of the neural network architecture is still arbitrary and not *physics-aware*. Note that using physics-based loss functions in the learning objective of ANN ensures that the search space of ANN models in the training phase is restricted to physically consistent options. However, once training is complete, there is no guarantee that the trained ANN model will be physically consistent on unseen test instances. This is because conventional ANN architectures do not have hard-coded physics-based

Feed Workshop SIGKDD'19, August 2019, Anchorage, Alaska, USA

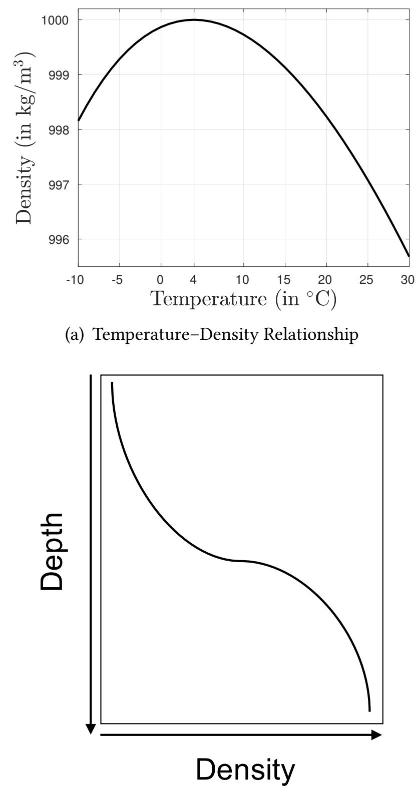
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

constraints in its design that guarantee it to be physically consistent over any unseen instance. Hence, minor perturbations of an ANN model, even if trained using physics-based loss functions, can lead to physical inconsistency. This can be a concern in scientific applications where it is important to quantify the uncertainty of model predictions, and every model prediction is required to be physically consistent, as described in the following.

Uncertainty quantification (UQ) is immensely crucial for model evaluation in a number of scientific and engineering applications, where rather than having point estimates of the target variable as outputs, it is preferred to have a distribution of the possible values of the target variable. In the context of neural networks, one way to perform UQ is by using dropout [10] on the trained neural network weights in the testing phase, to produce Monte Carlo simulations of the target variable for every test instance (a different simulation for every dropout version of the ANN model) [3]. This use of "dropout during testing" can be shown to approximate the Bayesian approach of assuming priors on the network weights and computing posterior estimates of the target variable [2]. Note that every dropout network represents a slightly perturbed version of the trained ANN model. Ideally, we would want every dropout network to produce physically consistent simulations of the target variable, so that the UQ analysis is physically meaningful. However, if we use conventional ANN architectures, we can easily obtain dropout networks that produce physically inconsistent solutions, even after being trained with physics-based loss functions.

In this paper, we explore a novel direction of research in theory-guided data science to develop *physics-aware architectures* of artificial neural networks (ANNs), where scientific knowledge is baked in the construction of ANN models. Figure 1 describes the high-level idea of creating physics-aware ANN architectures for a supervised learning problem of mapping inputs X to outputs Y. Instead of learning a black-box neural network model that uses arbitrary combinations of hidden layers to map X to Y, we consider a physics-aware ANN architecture where a block of hidden layers first maps X to intermediate physical variables, Z, which can then be mapped to the target space Y using known physical principles. By hard-wiring physics in the design of neural networks and by introducing physical variables as intermediate terms in the learned mapping from X to Y, we are taking a step towards physical interpretability of neural network solutions. Furthermore, a physics-aware ANN architecture is designed to comply with physical principles regardless of small perturbations in the neural network weights, e.g, as is the case in dropout networks. Hence, physics-aware ANN architectures are amenable to performing UQ in scientific applications, without violating the underlying physics in any of the model simulations. We explore novel physics-aware ANN architectures for the problem of lake temperature modeling and present preliminary results of these architectures in modeling the temperature of water in Lake Mendota, Wisconsin.

The remainder of the paper is organized as follows. Section 2 provides a brief background of the problem of lake temperature modeling and existing work in the area of physics-guided neural networks. Section 3 describes our proposed physics-aware ANN architecture. Section 4 presents some preliminary results of our proposed architecture on Lake Mendota. Section 5 provides concluding remarks and discusses directions for future research.



(a) Temperature–Density Relationship

(b) Density–Depth Relationship

Figure 2: Plots of physical relationships between temperature, density, and depth of water that serve as the basis for introducing density prediction in Physics Aware Neural Network architecture.

2 BACKGROUND

2.1 Lake Temperature Modeling

We consider the problem of modeling the temperature of water in lakes as an illustrative application for developing physics-aware ANN architectures. The temperature of water in a lake is known to be an ecological "master factor" [7] that controls the growth, survival, and reproduction of fish (e.g., [9]). Accurate water temperatures (observed or modeled) are critical to understanding contemporary change, and for predicting future thermal for economically valuable fish.

Since observational data of water temperature at broad spatial scales is incomplete (or non-existent in some regions) high-quality temperature modeling is necessary. Of particular interest is the problem of modeling the temperature of water at a given depth[1], d, and on a certain time, t. This problem is referred to as 1D-modeling of temperature (depth being the single dimension). A number of physics-based models have been developed for studying lake temperature, e.g., the state-of-the-art general lake model (GLM).

2.2 Physics-guided Neural Networks

A recent line of work [4, 6] introduced the framework of physics-guided neural networks (PGNN) for the problem of lake temperature modeling. The basic motivation of PGNN is to use physics-based loss functions in the learning objective of ANN to ensure that the trained ANN model is generalizable as well as physically consistent. In particular, an ANN model can be termed physically consistent if its predictions of the target variable (i.e., temperature) is compliant with known physical relationships between the target variable and other physical variables (e.g., density, depth, and thermal energy). For example, Karpatne et al. [6] used a key physical relationship between temperature, density, and depth, which comprises of two main components.

First, the density of water, ρ, shares a known physical relationship with the temperature of water, Y, which can be described using the following quadratic equation [8]:

$$\rho = 1000 \times \left(1 - \frac{(Y + 288.9414) \times (Y - 3.9863)^2}{508929.2 \times (Y + 68.12963)}\right) \quad (1)$$

Figure 2(a) shows a plot of this relationship between temperature and density, where we can see that water is maximally dense at 4°Celsius. Given the temperature predictions of a model, $\hat{Y}[d, t]$, at depth, d, and time-step, t, we can use Equation 1 to compute the corresponding density prediction, $\hat{\rho}[d, t]$.

Second, the density of water monotonically increases with depth as shown in the example plot of Figure 2(b), since denser water is heavier and goes down to the bottom of the lake. Formally, the density of water at two different depths, d_1 and d_2, on the same time-step, t, are related to each other in the following manner:

$$\rho[d_1, t] - \rho[d_2, t] \leq 0 \quad \text{if } d_1 < d_2. \quad (2)$$

Given the knowledge of this physical relationship, one can construct a physics-based loss function as follows. On any pair of consecutive depth values, d_i and d_{i+1} ($d_i < d_{i+1}$), we can compute the difference in the density estimates of a model on time-step t as

$$\Delta[i, t] = \hat{\rho}[d_i, t] - \hat{\rho}[d_{i+1}, t] \quad (3)$$

A positive value of $\Delta[i, t]$ can be viewed as a violation of the physics-based equation 2 on depth d_i and time t. Hence, a simple approach to construct the physics-based loss function is to consider the mean of all physical violations across all consecutive depth-pairs and time-steps as follows:

$$PHY.Loss(\hat{Y}) = \frac{1}{n_t(n_d - 1)} \sum_{t=1}^{n_t} \sum_{i=1}^{n_d - 1} \text{ReLU}(\Delta[i, t]). \quad (4)$$

Jia et al. [4] extended the framework of PGNN to work with time-based LSTM architectures, where the variations in tempreature across time at a given depth was modeled as a sequence output of an LSTM. They also introduced additional physical constraints of energy conservation in the design of physics-based loss functions. Building on these previous efforts, in this work we consider the physical relationships between temperature, density, and depth (shown in Figure 2) to construct physics-aware neural network architectures for lake temperature modeling.

[1]Depth is measured in the direction from the surface of the water to the lake bottom.

3 PROPOSED APPROACH

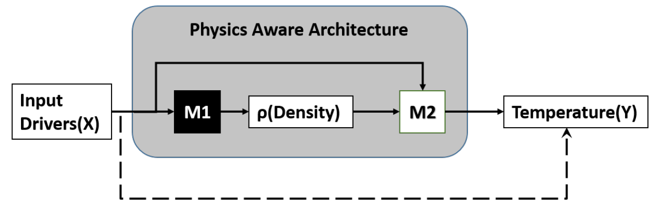


Figure 3: A schematic overview of our proposed physics-aware architecture for lake temperature modeling.

Figure 3 provides a schematic overview of our physics-aware neural network architecture for lake temperature modeling. The proposed physics-aware architecture comprises of two major components. First, we build a neural network model, M_1, that maps the input drivers X to an intermediate physical variable, density (ρ), which we know exhibits a direct physical relationship with the target variable, temperature (Y). Second, we map predictions of density to temperature using model M_2. Both these components are briefly described in thee following.

3.1 Density prediction model

The lake temperature modeling problem showcases two different dimensions of sequences, one being in the time dimension and the other being in the depth dimension. One can construct LSTM models to represent either of these two types of sequences for our problem. However, we consider depth-based LSTM sequences to represent lake temperature values in our study, since depth is a natural choice of dimension to encode physical relationships between density and depth. In particular, at a given time t, the recurrence relation in the depth-based LSTM architecture ensures that the density at depth d depends on the density at depth $d - 1$, which we know to satisfy a monotonically increasing relationship. More formally,

$$\rho_{d,t} = f(x_{d,t}, \rho_{d-1,t}) \quad (5)$$

where $x_{d,t}$ and $\rho_{d,t}$ represents the input drivers and the density at depth d and time t, respectively. The input drivers X consist of: day of year, depth, air temperature, short wave radiation, long wave radiation, relative humidity, wind speed, rain, growing degree days, is freezing and is snowing. All of these features except the depth are only measured at the surface of the lake, and thus remain constant for a particular time t across all depths. Figure 4 provides an overview of the LSTM architecture, where the input drivers are first passed through LSTM blocks to extract high-dimensional temporal features, which are then feed into two dense layers followed by a regression node at the output layer to get the density predictions at each depth and time-step. The monotonic physical relationship between density and depth (i.e. the density of the water in the lake gradually increases with depth) can be exploited in the LSTM architecture in a subsequent study.

3.2 Density to Temperature model

The second model focuses on mapping density estimates of M_1 to estimates of the target variable, temperature. Since the physical relationship between temperature and density is quadratic in nature

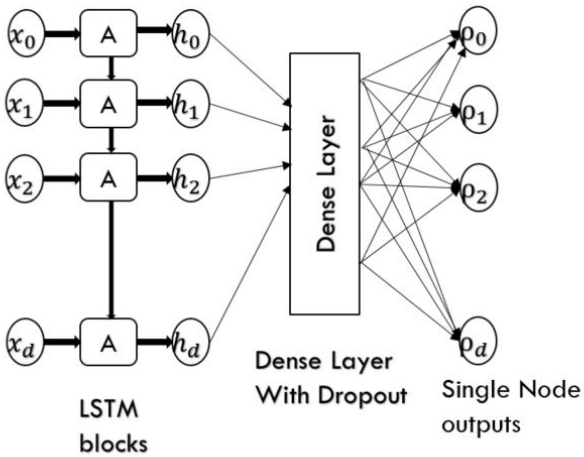


Figure 4: An overview of the depth-based LSTM model for density prediction

(see Equation 1 and Figure 2(a)), we can see that there are two distinct roots of temperature for every value of density, and hence, the mapping from density to temperature is non-unique. We thus use a data-driven model to select one of the two roots of the physical equation depending on the values of the input drivers, X. In general, if we want to transform density to temperature, we would need information about input drivers such as air temperature at the surface of the lake and the day of year to identify which one of the phases of temperature (less than 4 or greater than 4) should be chosen. As a simplification, we used the day of year, air temperature and the density to predict the corresponding temperature values using a basic feed-forward neural network model.

4 EVALUATION

4.1 Data

We consider Lake Mendota in Wisconsin, USA to demonstrate the results of our physics-aware ANN architecture. This is a reasonably large lake with a surface area of approximately 40 km^2 and shows sufficient dynamics in temperature profiles across depth over time. The overall data for Mendota comprised of 13,543 temperature observations from 30 April 1980 to 02 Nov 2015. For each observation, we used a set of 11 meteorological drivers as input variables, as used in previous studies [4, 6].

4.2 Experimental Design

Due to non-uniformity in observations across depth and time, we pre-processed the data to create uniform depth samples for each timestep, and any missing depth observations were masked out. Using this procedure, out of approximately 35 years of observations, we only had 662 observable dates. We partitioned these dates into two contiguous time windows to be used for training and testing, such that there is no temporal auto-correlation between the training and test sets. In particular, we considered first 80% observations starting from 30 April 1980 for training, and used the remainder of the data for testing. For the density prediction model, both the input and the output variables were normalized to zero mean and unit standard deviation. The density to temperature conversion model on the other hand is trained to learn the mapping from the air temperature at the surface of the lake, day of year and the GLM densities (obtained by converting the GLM temperature predictions into density using Equation 1) to their corresponding GLM temperature predictions.

The Keras package [1] using Tensorflow Backend was used for implementing the neural networks described in this paper. The Adam optimiser was used to optimise the model parameters of the neural network. For the density prediction model, a batch size of 100 with maximum number of epochs set to be 1000, whereas for the density to temperature conversion model, a batch size of 500 with maximum number of epochs of 10,000 was set. To avoid overfitting of each of the two models, early stopping was implemented where 10% of the training data was reserved for validation, and a patience value of 300 and 3000 was set for the two models M1 and M2, respectively. Gradient clipping was also performed to avoid the problem of exploding gradients. The weights of the neural network were also randomly initialised between 0 and 1. To obtain uncertainty estimates, the dropout method was used during the testing phase [3]. Randomly dropping out a small fraction (5-10%) of model parameters creates a small perturbation which results in a slightly different model output for the same input, thus resulting in uncertainty quantification.

4.3 Evaluation Metrics

We consider the following evaluation metrics for comparing the different algorithms proposed in this paper:

- **RMSE**: We use the root mean square error(RMSE) of a model on the test set as a metric of generalizability of the model. The lower the value of the RMSE, the better is its generalizability.
- **Physical Inconsistency**: The physical inconsistency is defined as the fraction of time-steps where the model makes physically inconsistent solutions. Thus, the physical inconsistency computation requires pairwise comparisons of consecutive depth values to check for violation of the density-depth relationship stated in Equation 2. We use a tolerance value of 10^{-4} kg/m^3 to decide if a difference in density across consecutive depths is physically inconsistent or not.
  - **Physical inconsistency without Dropout**: Physical inconsistency without Dropout is defined as the physical inconsistency of the temperature profiles without using dropout to generate the uncertainty estimates.
  - **Average Physical inconsistency**: Average Physical inconsistency is defined as the average of the physical inconsistencies of each of the Monte Carlo predictions of temperature obtained using dropout networks during the testing phase.

4.4 Baselines

Our proposed physics-aware architecture for lake temperature modeling will be referred to as the ρ-LSTM in the remainder of this paper. We compare our ρ-LSTM with the following baseline methods:

	RMSE	Phy. Incon. w/o Dropout	Avg. Phy. Incon.
GLM	2.73	0	-
ρ-LSTM	2.3 ± 0.12	0	0.914±0.024
T-NN	2.305±0.10	0.520±0.041	-
T-LSTM	1.65±0.07	0.290±0.026	1

Table 1: Comparison of baseline methods with ρ-LSTM.

- **GLM**: The General Lake Model(GLM), a state of the art physics model available for Lake Temperature modelling was used as a baseline for comparing our proposed model.
- **T-NN**: The T-NN represents the temperature-based model which directly maps the input space X to the output temperature space Y using a basic feed-forward neural network architecture. We used a neural network with 2 layers and 20 neurons at each layer for this baseline method.
- **T-LSTM**: The architecture of this temperature-based LSTM (or T-LSTM) is similar to the one we have used for density prediction (model M_1). The only difference is that rather than predicting density, this model directly predicts the temperature.

4.5 Preliminary Results

Table 1 provides a comparative summary of performance of the baseline methods and ρ-LSTM. If we look at the RMSE values, it is evident that the proposed ρ-LSTM model performs similar to a T-NN model, but the T-LSTM model outperforms by quite a margin. To understand this, we need to look into the architecture of the ρ-LSTM, which comprises of two primary components as described in Section 3: the density prediction model and the density to temperature model. The RMSE of the density prediction model is around 0.22×10^{-3} kg/m^3 (which is quite small), whereas the RMSE of the density to temperature model is around 0.53°. Hence, the RMSE bottleneck for the ρ-LSTM model is the high RMSE of M_2. Indeed, by using better approaches for inverting the physical relationship between density and temperature, we can achieve lower RMSE values.

To further understand the variations in RMSE, Figure 5 shows depth-wise RMSE of our ρ-LSTM model at different depth values. We can see that there is an increase in RMSE in mid-depth ranges, followed by a sharp decrease in RMSE at higher depths, which can be explained as follows. For a lake, a thermocline is defined as a thin layer of water showing steep temperature gradients, which prevents the mixing of the waters separated by the layer. Just below the thermocline, the temperatures might vary up to 9 degrees with just one meter increase in depth. The thermocline shifts over the seasons and shows one of the most complex dynamics to capture in a lake temperature model. The typical depth of thermocline for Lake Mendota is around 8-10 meters, which justifies the high RMSE just below that range. Also, the rapid decrease in RMSE in the greater depths could be explained by the low rate of mixing of waters at the bottom of the lake due to the presence of the thermocline. This can also be observed from figure 6, where the standard deviation of temperature across time is very small at the bottom of the lake.

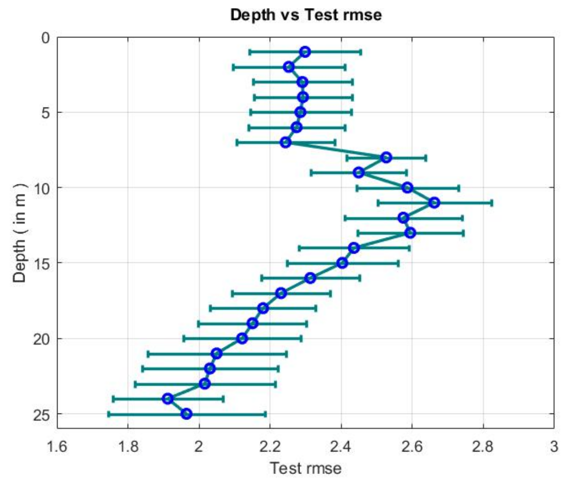


Figure 5: Depth vs Test RMSE of the ρ-LSTM

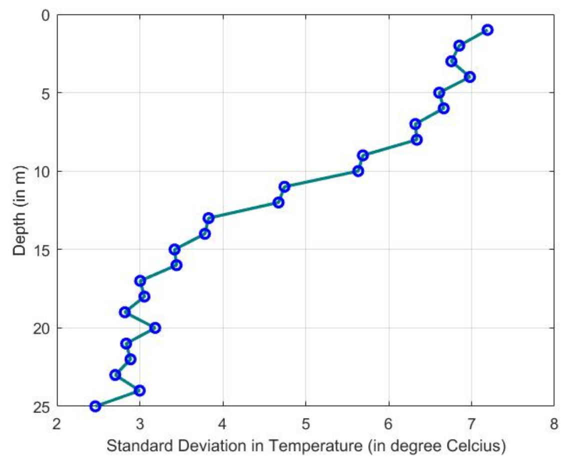
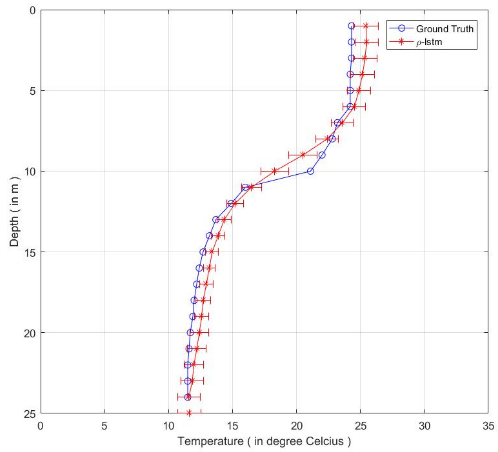


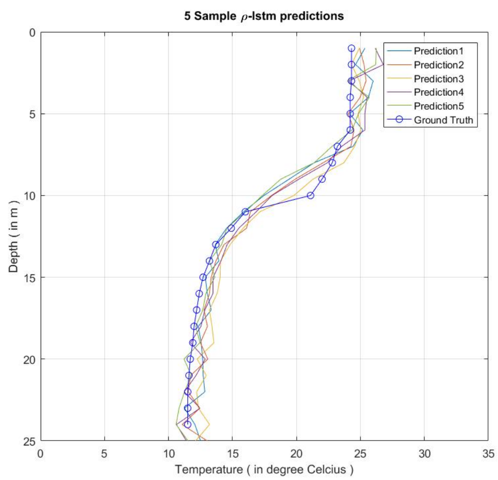
Figure 6: Depth vs Standard Deviation of Temperature across time of the ρ-LSTM

We can also see from Table 1 that ρ-LSTM has significantly lower physical inconsistency without dropout as compared to baseline methods, but when it comes to evaluation of each of the individual Monte Carlo dropout predictions, both ρ-LSTM and baseline methods have a very high average physical inconsistency. However, ρ-LSTM still shows lower average physical inconsistency than the baseline methods, thus demonstrating the advantage of using a physics-aware architecture that predicts an intermediate quantity, density, and then maps it to the target variable, temperature, rather mapping the target variable directly. Further, ρ-LSTM provides a natural way to hard-code physical relationships between density and depth in the LSTM architecture, which we will investigate in our future study.

Figure 7(a) shows the temperature profiles of ρ-LSTM by plotting the means of the individual dropout predictions with their corresponding standard deviations as uncertainty estimates along



(a) Mean and standard deviation of ρ-LSTM dropout profiles



(b) 5 out of 100 individual dropout profiles for ρ-LSTM

Figure 7: Visualization of dropout temperature profiles of ρ-LSTMs on a sample date.

with the ground truth. Figure 7(b) depicts 5 of the 100 sample ρ-LSTM predictions produced using dropout during testing for ease of visualization. It can be seen that the means of the individual profiles are physically consistent but each individual profile is quite physically inconsistent. In our future study, we plan to address by hard-coding the monotonic constraint between density and depth in the design of density prediction model, M_1.

5 CONCLUSION AND FUTURE WORK

This paper presented a novel direction of building physics-aware neural network architectures, where rather than directly mapping the input drivers X to the output space Y, we predict intermediate physical variables Z, such that the output Y is physically dependent on Z. This not only improves the physical consistency of the solution but also increases the interpretability of the model. Any physical relationship between intermediate variable Z and the input X can also be hard-wired into the model to force it to comply with the constraint.

There are a number of directions we will explore in future research. First, we want to have a monotonically constrained depth-based LSTM architecture for the density prediction model, such that the physical relationship between density and depth is hard-coded into the architecture. This would allow us to obtain physically consistent outputs even for each of the individual profiles. Second, we want to modify the density to temperature neural network model to a fully physics-based model. This would allow us to remove the bottleneck which is causing the higher RMSE in our model predictions. Third, we will incorporate temporal relationships in our depth-based LSTM architecture to further improve the performance of the model. Fourth, we will explore alternate ways of training the two component models jointly, rather than training them individually. Finally, we will explore other scientific and engineering applications where physical relationships can be hard-coded in novel physics-aware ANN architectures.

REFERENCES

[1] FranÃğois Chollet. 2015. keras. https://github.com/fchollet/keras.
[2] Meire Fortunato, Charles Blundell, and Oriol Vinyals. 2017. Bayesian recurrent neural networks. *arXiv preprint arXiv:1704.02798* (2017).
[3] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*. 1050–1059.
[4] Xiaowei Jia, Jared Willard, Anuj Karpatne, Jordan Read, Jacob Zwart, Michael Steinbach, and Vipin Kumar. 2019. Physics Guided RNNs for Modeling Dynamical Systems: A Case Study in Simulating Lake Temperature Profiles. In *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, 558–566.
[5] Anuj Karpatne, Gowtham Atluri, James H Faghmous, Michael Steinbach, Arindam Banerjee, Auroop Ganguly, Shashi Shekhar, Nagiza Samatova, and Vipin Kumar. 2017. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering* 29, 10 (2017), 2318–2331.
[6] Anuj Karpatne, William Watkins, Jordan Read, and Vipin Kumar. 2017. Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv preprint arXiv:1710.11431* (2017).
[7] John J Magnuson, Larry B Crowder, and Patricia A Medvick. 1979. Temperature as an ecological resource. *American Zoologist* 19, 1 (1979), 331–343.
[8] James L Martin and Steven C McCutcheon. 1998. *Hydrodynamics and transport for water quality modeling*. CRC Press.
[9] James J Roberts, Kurt D Fausch, Douglas P Peterson, and Mevin B Hooten. 2013. Fragmentation and thermal risks from climate change interact to affect persistence of native trout in the Colorado River basin. *Global Change Biology* 19, 5 (2013), 1383–1398.
[10] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.