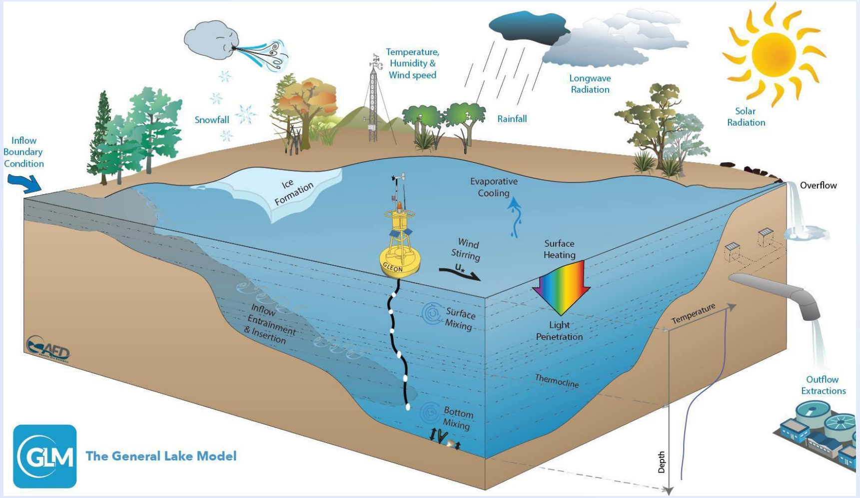
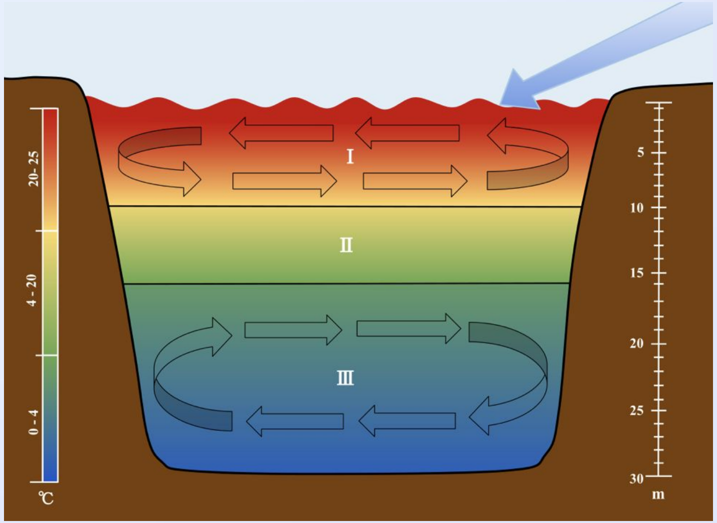
Modelling Lake Thermal Stratification

Nan Zhang, George Lu, Charlie Sturr, Bryn Stecher

Project Overview

- In this project, we are interested in understanding the utility of all the moving parts of the proposed process guided deep learning algorithms in the paper and **evaluating possible improvements** for predicting lake water temperatures.





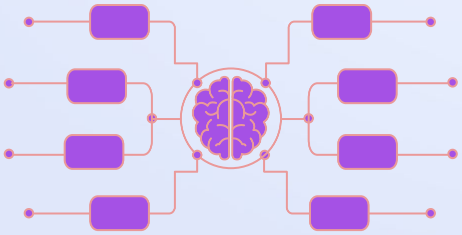
- Image source: https://aed.see.uwa.edu.au/research/models/glm/

Methodology

01

Optimize the Model

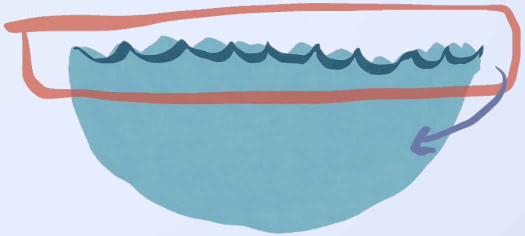
We were testing how the data runs with different algorithms with/without dynamic learning rate



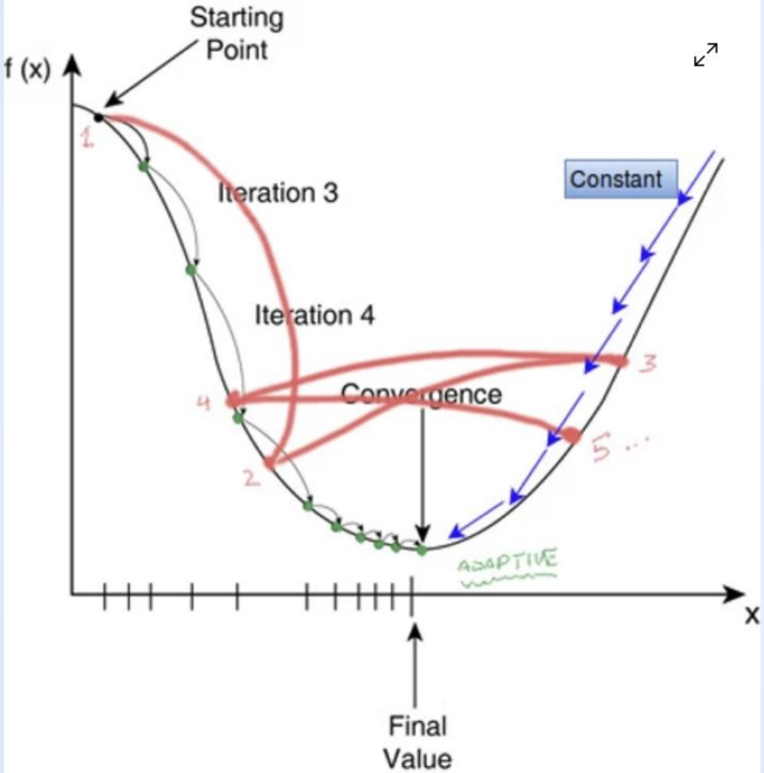
02

Use Less Data

To save memory and increase the speed, we looked at only a top section of the lake depth



Dynamic (Adaptive) v.s. Fixed Learning Rate



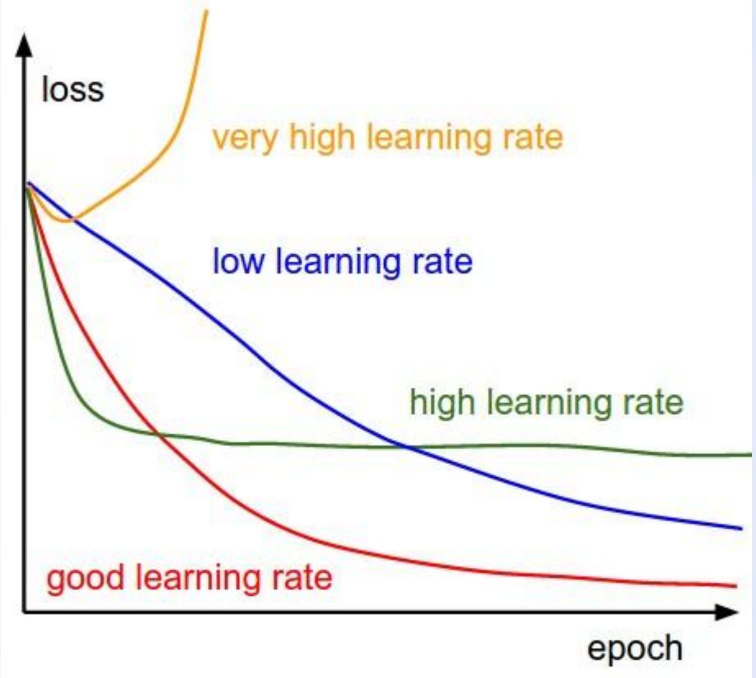


Image source: https://amirhessam88.github.io/adaptive-learning/

Image source: https://snlpatel0012134.wixsite.com/thinking-machine/single-post/artificial-neural-network-effect-of-adaptive-learning-rate

Note: results show the average across 10 test runs of each scheme

Optimizer

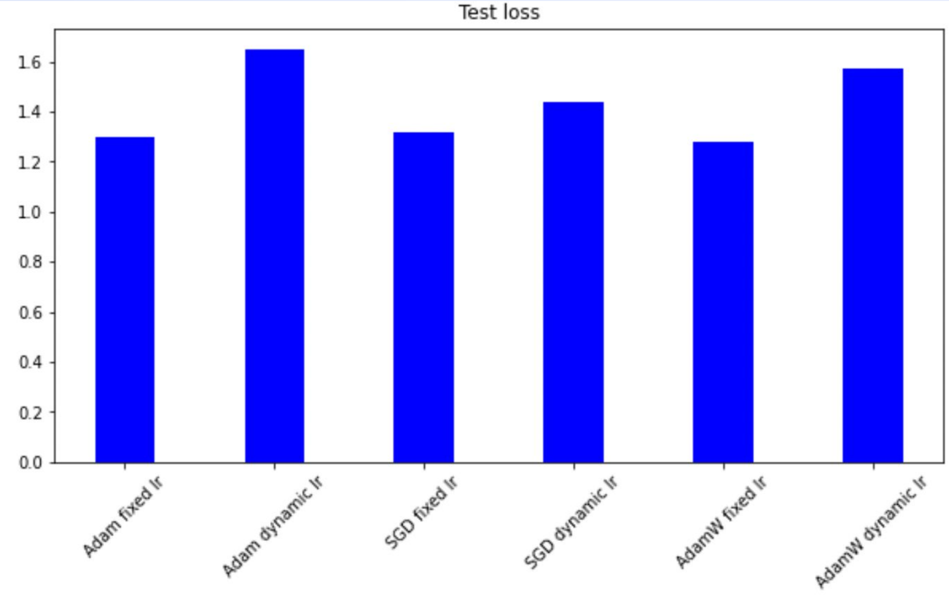
*“**Optimizers** can be explained as a mathematical function to modify the weights of the network given the gradients and additional information, depending on the formulation of the optimizer. Optimizers are built upon the idea of gradient descent, the greedy approach of iteratively decreasing the loss function by following the gradient.”*

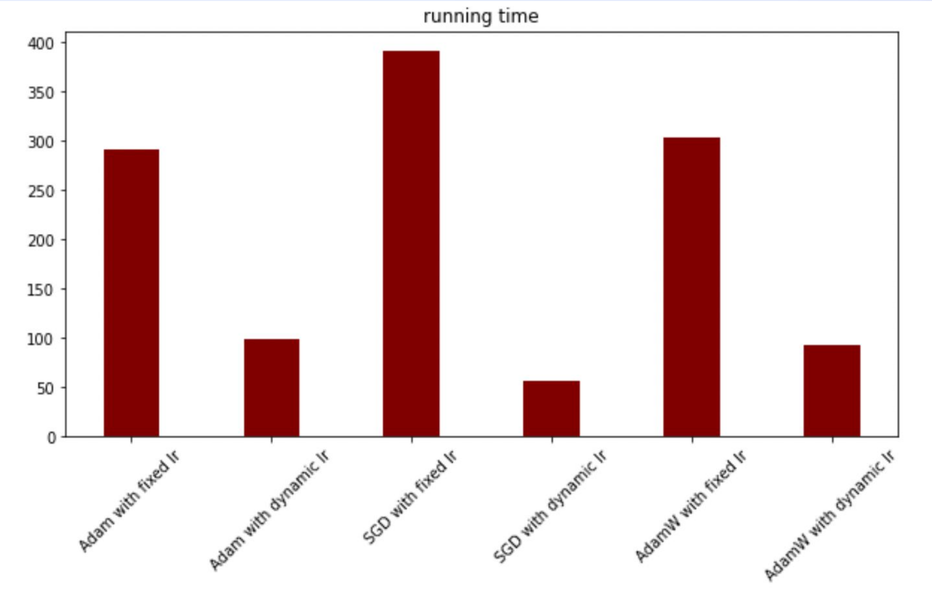
Here we tested the performance of three widely used optimize algorithms with/without dynamic learning rate

- Adam Algorithm
- AdamW Algorithm
- SGD (Stochastic Gradient Descent) Algorithm

Note: results show the average across 10 test runs of each scheme

Part I: Testing Different Optimizers

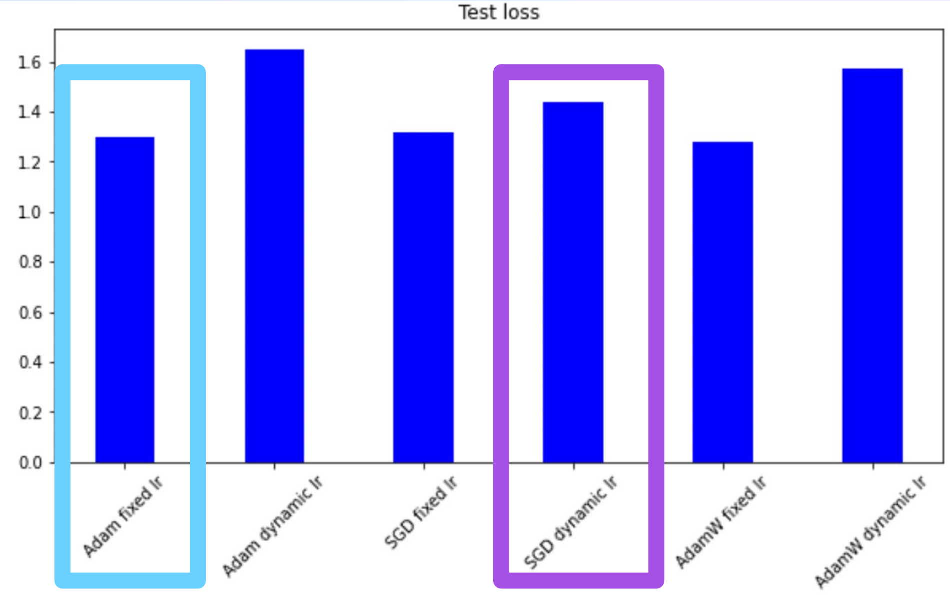


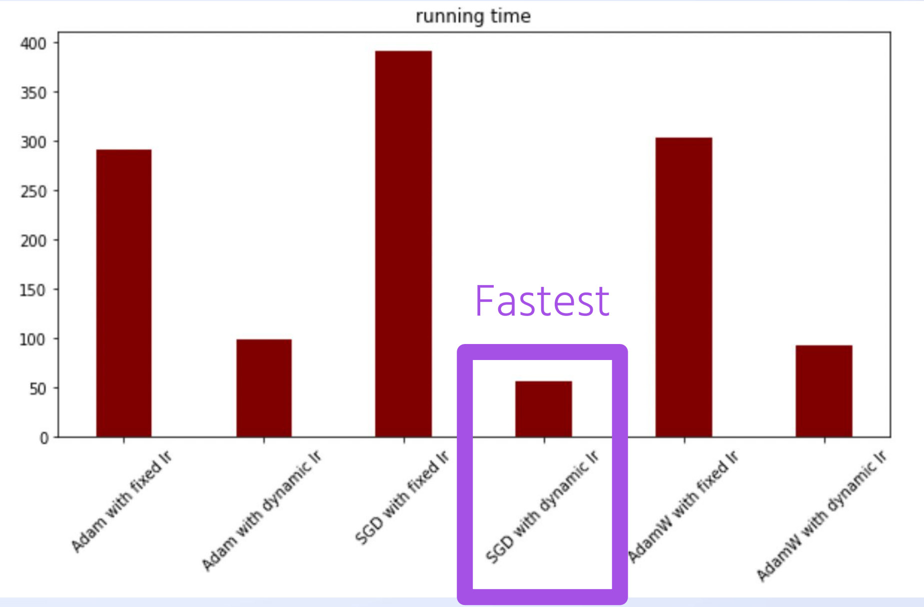


lr = Learning Rate

Note: results show the average across 10 test runs of each scheme

Part I: Testing Different Optimizers

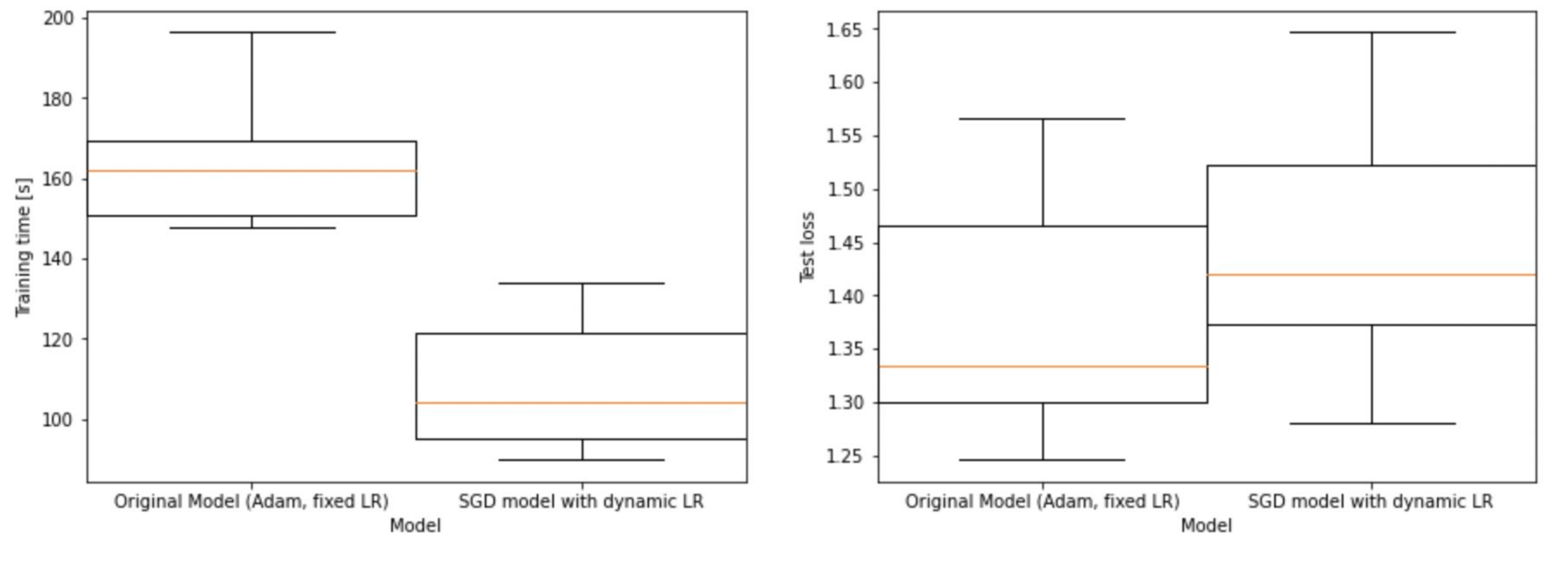




lr = Learning Rate

Note: results show the average across 10 test runs of each scheme

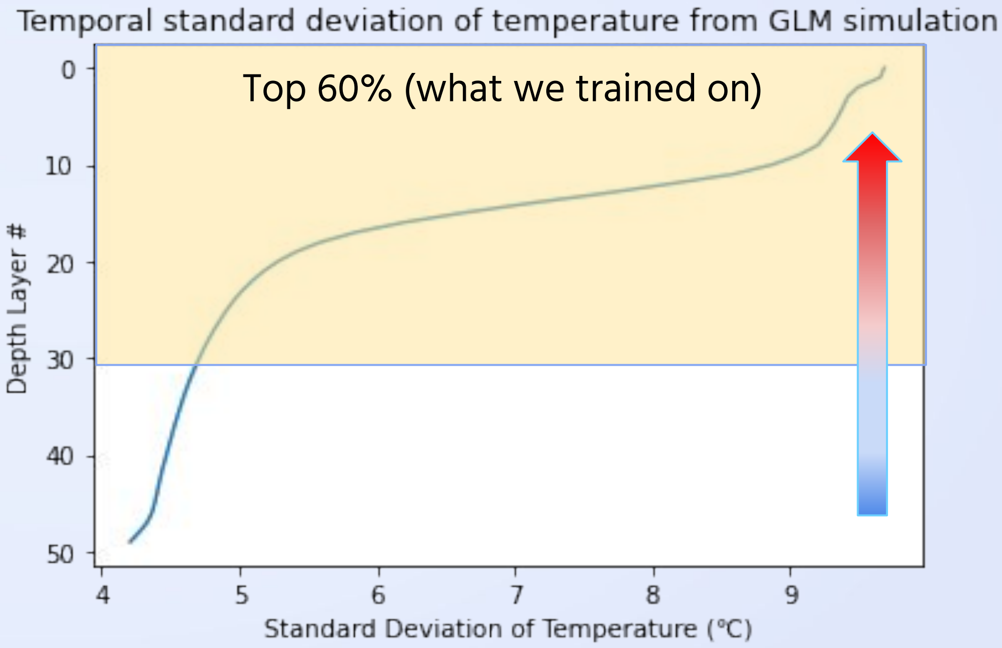
Part I: Results



Although the dynamic learning rate slightly increases the loss on occasion, it drastically reduces the training time consistently

Part II: Altering Lake Depth

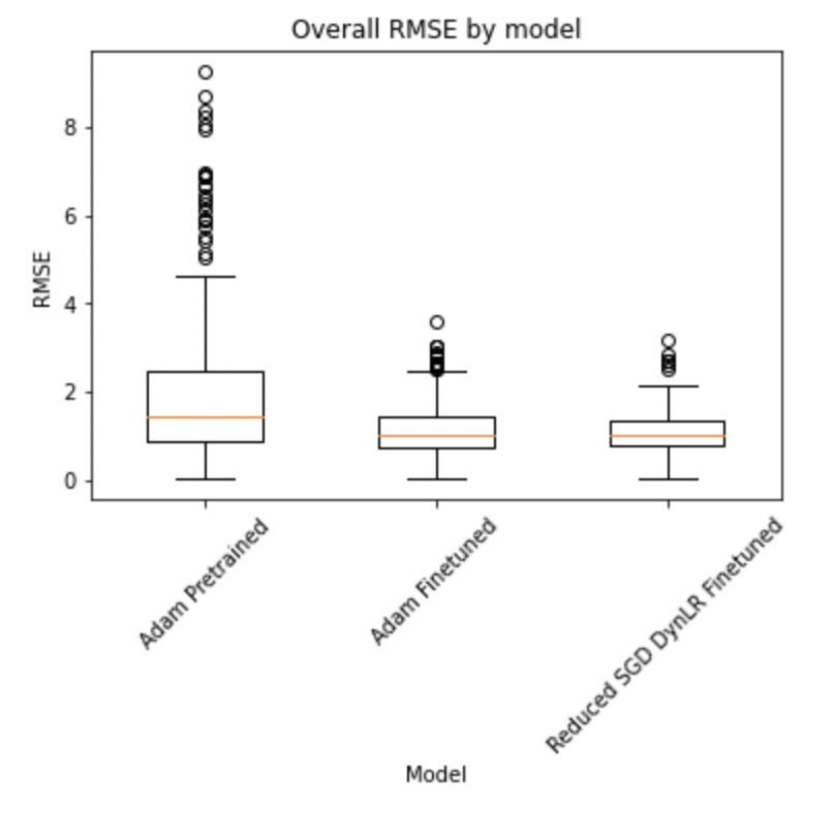
One factor we wondered about is if the **full depth profile** was needed to train the data. Some initial data analysis reveals that the **temperatures at the bottom of Lake Mendota vary a lot less than the surface temperatures**. We visualize this with our simulated temperatures.



Part II: Results

The results generally returned to be about the same RMSE (root-mean-square deviation) as the original one

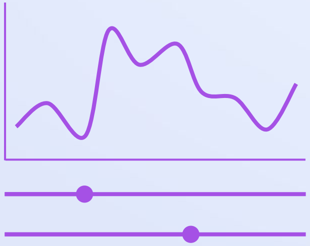
It is also consistently faster comparing with the situation that trained with the whole depth profile (timings will vary with each run and machine however)



Summary

Optimizer

SGD Algorithm +
Dynamic Learning Rate



+

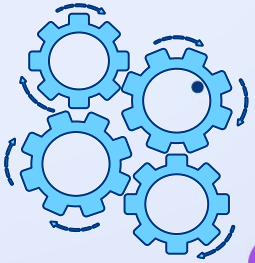
Reduced Depth Sampling

Does not impact the results too much, while allowing the program use less memory and run faster

=

Faster Speed & Less Memory

Speed is significantly faster than the original model



Reference

- Read, J. S., Jia, X., Willard, J., Appling, A. P., Zwart, J. A., Oliver, S. K., ... & Kumar, V. (2019). Process‐guided deep learning predictions of lake water temperature.

- Jia, X., Willard, J., Karpatne, A., Read, J., Zwart, J., Steinbach, M., & Kumar, V. (2019, May). Physics guided RNNs for modeling dynamical systems: A case study in simulating lake temperature profiles.

- Jia, X., Willard, J., Karpatne, A., Read, J. S., Zwart, J. A., Steinbach, M., & Kumar, V. (2021). Physics-guided machine learning for scientific discovery: An application in simulating lake temperature profiles.

- https://medium.com/geekculture/a-2021-guide-to-improving-cnns-optimizers-adam-vs-sgd-495848ac6008