The role of machine learning in ice sheet modeling

Presented on March 24th, 2022 by Kyle Blum

The goals this presentation:

- Discuss how we calibrate our ice sheet model
- Give a brief overview of why machine learning is a good way to solve this problem
- Describe how our neural net learns PISM
- Give a few details on the implementation of this neural net with pytorch

What we're looking at here is a classic inverse problem

$$G(m) = d$$

What we’re looking at here is a classic inverse problem

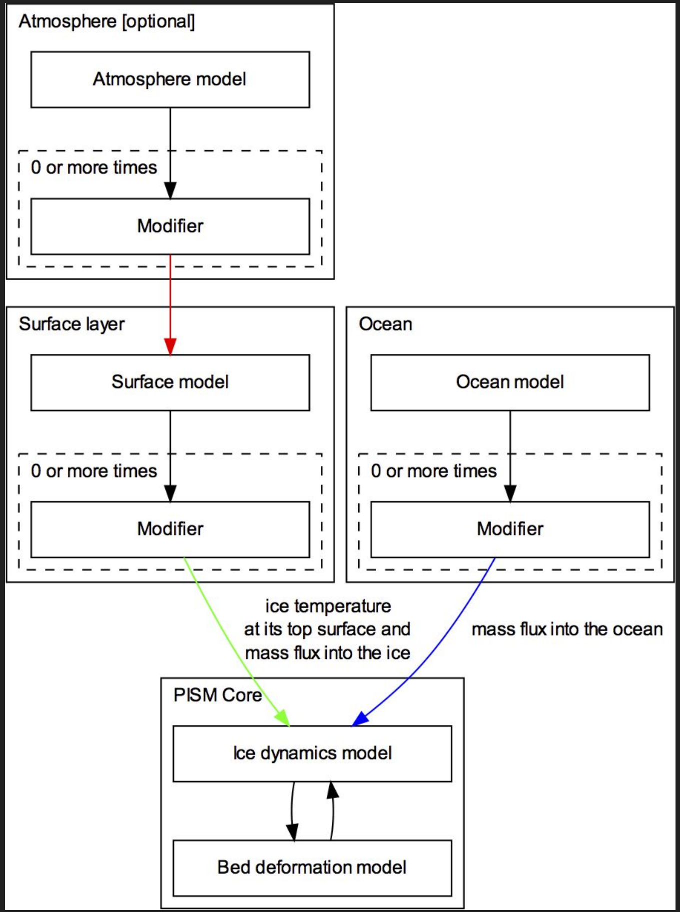
G = PISM

m = ice dynamics parms

d = velocity fields

Ice sheet modeling is expensive

G =

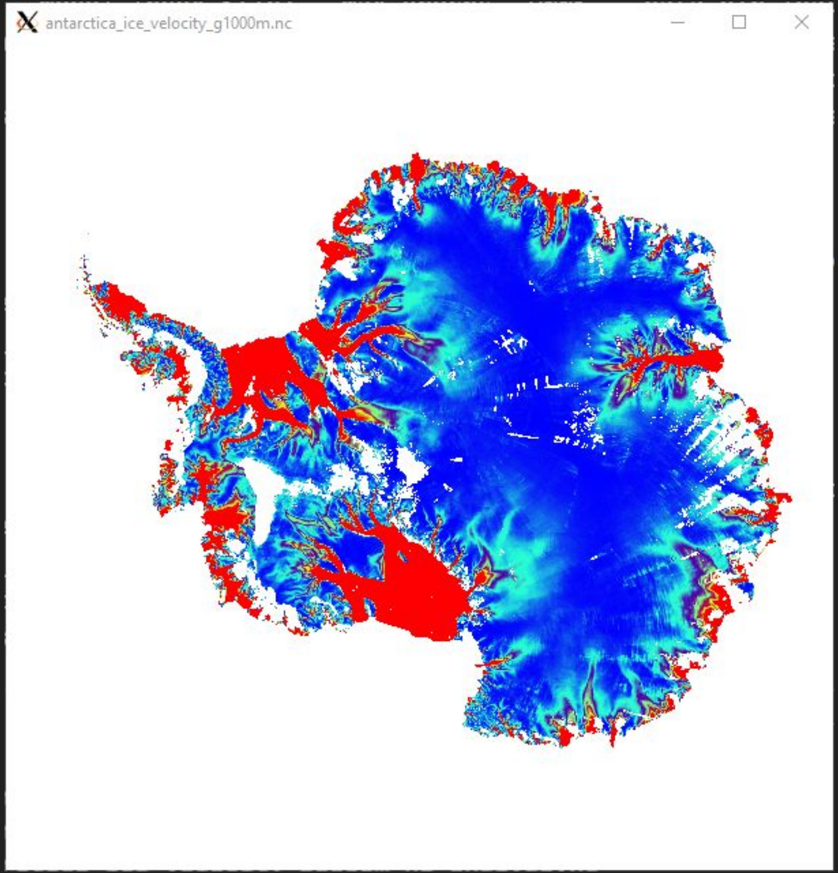


There are many (>500) model parameters that dictate our model

$$m = m_0 + m_1 + \ldots + m_{510}$$

$$m_{flow} = m_0 + m_1 + \ldots + m_7$$

We can represent surface flow with SVD

$d =$ 

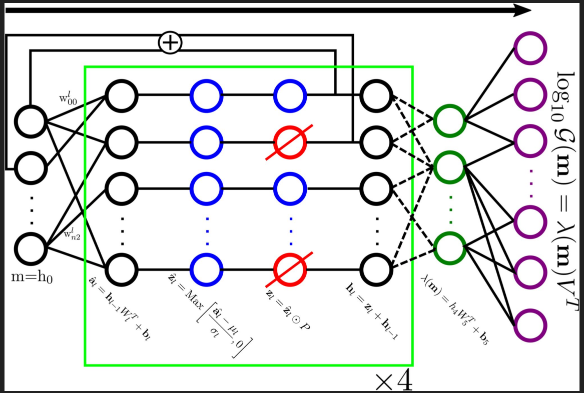
$= USV^T$

Ice sheet modeling is expensive

G = PISM

PARALLEL ICE SHEET MODEL

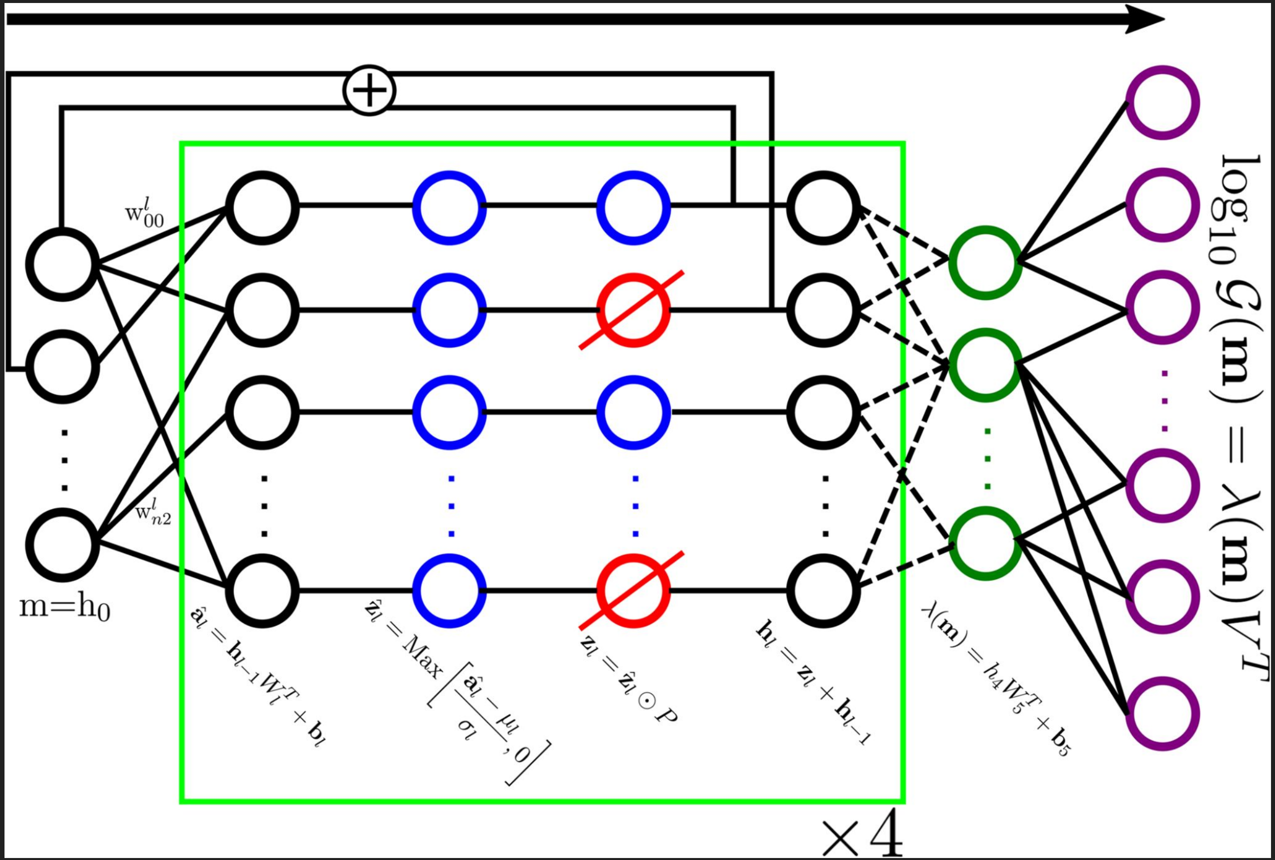
Model calibration doesn't have to be as expensive



= F ≅ G =

PISM
PARALLEL ICE SHEET MODEL

We implement an L=5 neural network with 128 nodes per layer



$$\hat{\mathbf{a}}_l = \mathbf{h}_{l-1} W_l^T + \mathbf{b}_l,$$

$$\hat{\mathbf{a}}_l = \mathbf{h}_{l-1} W_l^T + \mathbf{b}_l,$$

$$\mathbf{a}_l = \alpha_l \frac{\hat{\mathbf{a}}_l - \mu_l}{\sigma_l} + \beta_l,$$

$$\hat{\mathbf{a}}_l = \mathbf{h}_{l-1} W_l^T + \mathbf{b}_l,$$

$$\mathbf{a}_l = \alpha_l \frac{\hat{\mathbf{a}}_l - \mu_l}{\sigma_l} + \beta_l,$$

$$\hat{\mathbf{z}}_l = \mathrm{ReLU}(\mathbf{a}_l),$$

$$\hat{\mathbf{a}}_l = \mathbf{h}_{l-1} W_l^T + \mathbf{b}_l,$$

$$\mathbf{a}_l = \alpha_l \frac{\hat{\mathbf{a}}_l - \mu_l}{\sigma_l} + \beta_l,$$

$$\hat{\mathbf{z}}_l = \mathrm{ReLU}(\mathbf{a}_l),$$

$$\mathbf{z}_l = \hat{\mathbf{z}}_l \odot R,$$

$$\hat{\mathbf{a}}_l = \mathbf{h}_{l-1} W_l^T + \mathbf{b}_l,$$

$$\mathbf{a}_l = \alpha_l \frac{\hat{\mathbf{a}}_l - \mu_l}{\sigma_l} + \beta_l,$$

$$\hat{\mathbf{z}}_l = \mathrm{ReLU}(\mathbf{a}_l),$$

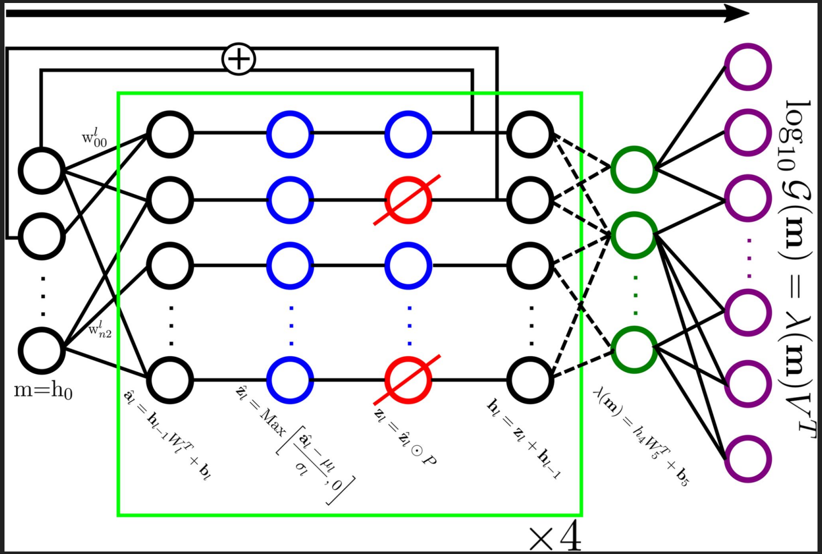
$$\mathbf{z}_l = \hat{\mathbf{z}}_l \odot R,$$

$$\mathbf{h}_l = \mathbf{z}_l + \mathbf{h}_{l-1},$$

F is the approximate map from m_{flow} to surface ice speeds

F

input: m_{flow}



output: d=speeds

To accrue training data we run the high fidelity model forward a bunch of times

$$G(m) = d_{target}$$

To train our surrogate model, we minimize this objective

$$I(\theta) \propto \sum_{i=1}^{m} [F(m_{flow,i}, \theta) - G(m_{flow,i})]^2$$

Let’s pop open the hood

PyTorch Lightning

Bibliography

Brinkerhoff, Douglas, et al. “Constraining Subglacial Processes from Surface Velocity Observations Using Surrogate-Based Bayesian Inference.” Journal of Glaciology, vol. 67, no. 263, 2021, pp. 385–403., https://doi.org/10.1017/jog.2020.112.

Khroulev, C., & the PISM Authors. (2022). PISM, a Parallel Ice Sheet Model. Zenodo. doi: 10.5281/zenodo.1199019

Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

I’m happy to take questions at this time