ECMWF Machine Learning Workshop 2022

Physics-Informed Learning of Aerosol Microphysics

Paula Harder

Mila · University of Oxford · Fraunhofer ITWM · Technische Universität Kaiserslautern

Content

Work done with:
Duncan-Watson Parris, Philip Stier, Janis Keuper, Dominik Strassel and Nico Gauger

Motivation

Data & Model

Physics-Informing

Results

Motivation

What are aerosols?

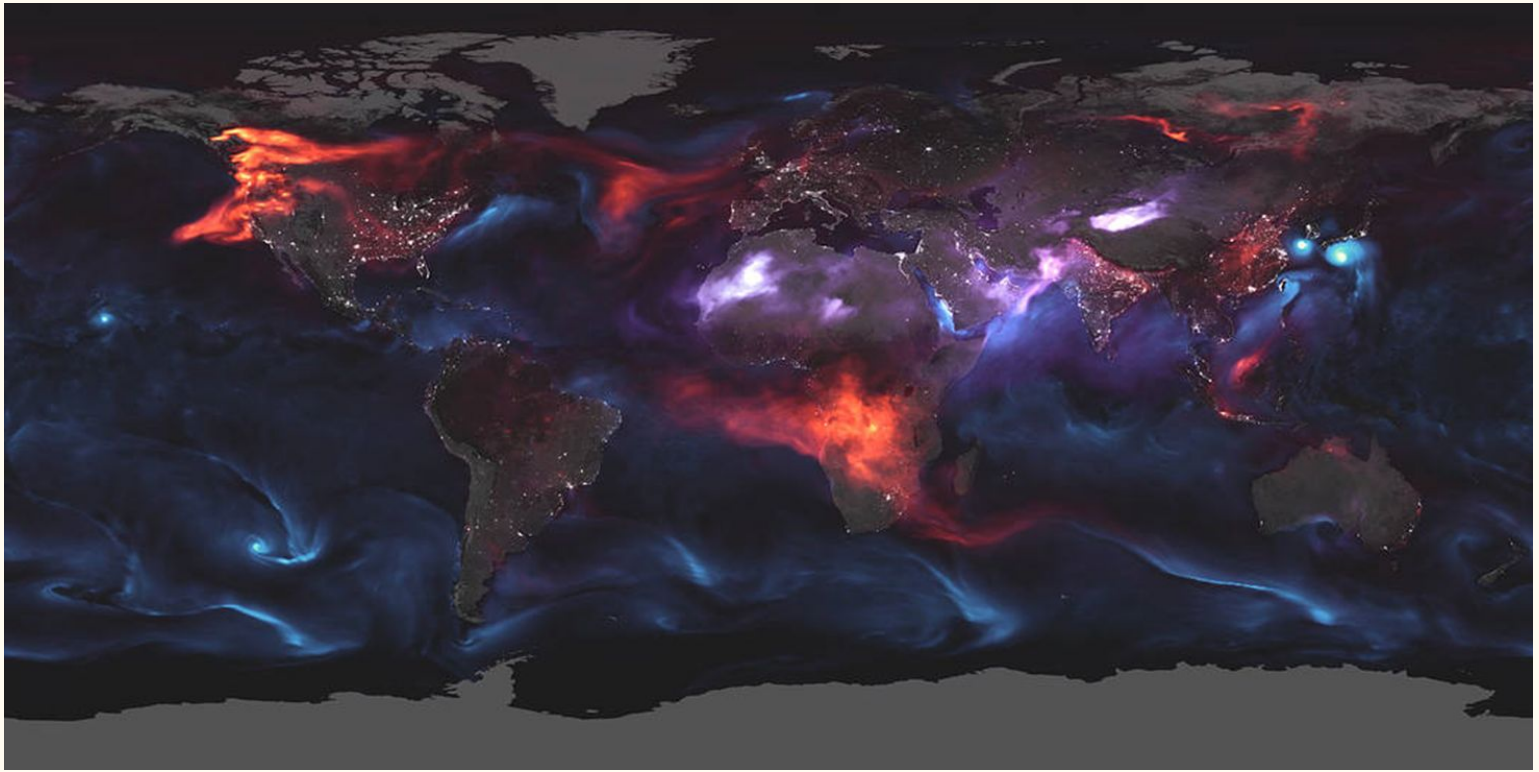
An aerosol is a suspension of fine solid particles or liquid droplets in air or another gas

Important atmospheric aerosols: Dust, sea salt, black carbon, organic carbon, sulphates

Natural and anthropogenic sources: Volcanic eruptions, forest fires, fossil fuel burning

Affecting the climate by aerosol-cloud and aerosol-radiation interactions

Aerosols



Red - black carbon, purple - dust, blue - sea salt

Image source: NASA

Motivation

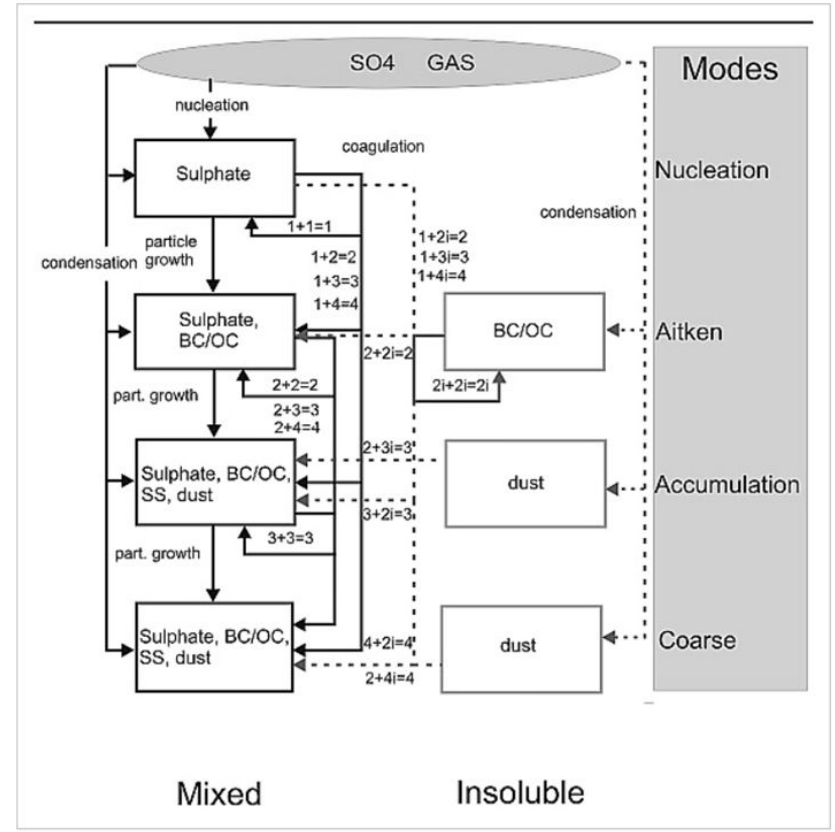
Most climate models do not include aerosols in detail and if they are included the computational costs do make long-term high-resolution runs impossible

Idea: Replace original computationally expensive model with cheap ML model = Emulation

The machine learning model is trained once offline and we benefit from fast inference time online

Data & Model

Aerosol Microphysics Model M7



Aerosol microphysics model by Vignati et al. 2004

Different aerosol types modeled with size bins

M7: An efficient size-resolved aerosol microphyiscs module for large-scale aerosol transport models, Vignatti et al. 2004

Data generation

Input data generated from runs of global aerosol-climate model ECHAM-HAM

Output data generated with M7 aerosol microphysics box model

~5.5M data points for training, ~3M for validation, ~3M for testing, each from different days spread out through the year

Data

Idea: Same input and output as traditional model, e.g. no spatial information

Predict one time step for fixed time step length, here 450s

32 input and 30 output variables

Input/Output: 26 values for masses and numbers of different aerosol types

Input only: Atmospheric variables, temperature, pressure, RH, ...

Output only: Water content of aerosols

Data challenges

Changes in variables are small to full variables → we model changes and evaluate on changes

Changes themselves are expontially distributed

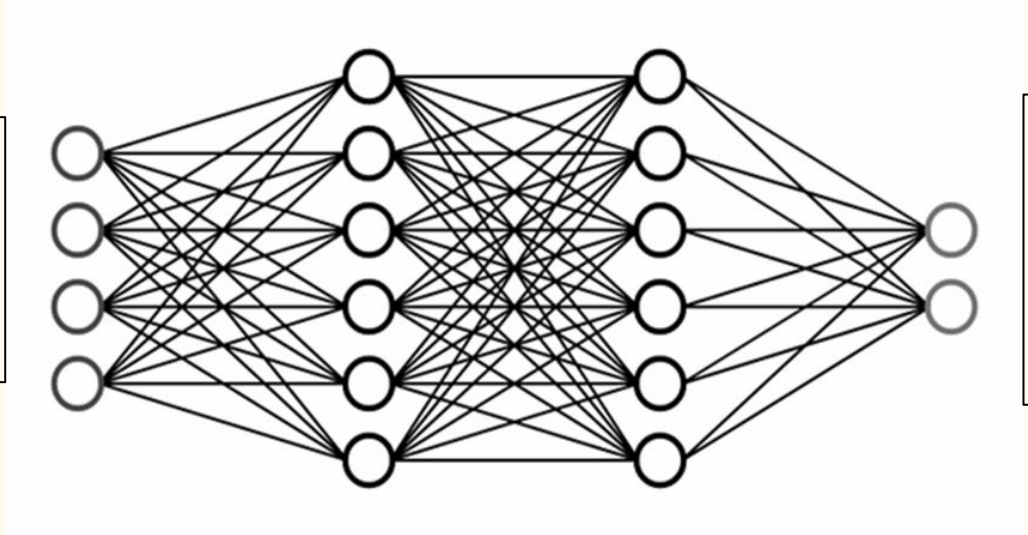
We compare using logarithmically transformed and just standardized variables

Log-transformation requires an additional network to classify whether a change is positive or negative

Network architecture

Neural network with 2 hidden layers, 256 nodes per hidden layer, ReLU activation

Inputs:
Temperature, RH, pressure,...
Aerosol masses
Aerosol number



Outputs:
Change in aerosol masses
Change in aerosol numbers
Water content

Physics-Constraining

Equality and inequality constraints

Let x be the NN's input and y the output, h, g (linear) functions

Equality constraints:

$$h(x, y) = 0$$

Inequality constraints:

$$g(x, y) \geqslant 0$$

See also: Enforcing Analytic Constraints in Neural Networks Emulating Physical Systems, Beucler et al. & DC3: A learning method with hard constraints, Donti et al.

Soft vs. Constraining

Let $\tilde{y}$ be the target

Soft constraining: The NN's loss L is extended by additional terms

$L(y,\tilde{y}) := ||y-\tilde{y}||^2 + \lambda$ Penalizer term

Hard constraining: Enforce constraints in additional network layer

Soft Constraints

$$L(x,y,\tilde{y}) := ||y-\tilde{y}||^2 + \lambda \underbrace{||h(x,y)||^2}_{\text{equality constraints}} + \mu \underbrace{||\text{ReLU}(-g(x,y))||^2}_{\text{inequality constraints}}$$

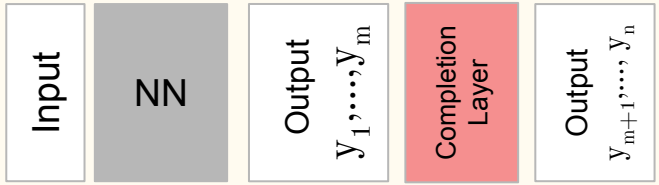
Equality constraints: Minimizing ||h(x,y)|| encourages the NN to output y, such that h(x,y) is close to 0

Inequality constraints: ReLU(z) = max(0, z), it sets the negative part to 0. Minimizing ||ReLU(-g(x,y))|| encourages the NN to output y, such that g(x,y) is positive

Hard Constraints

Equality constraints: Output partial set (y_1, ..., y_m) and then calculate (y_{m+1}, ..., y_n), such that $h(x,y)=0$

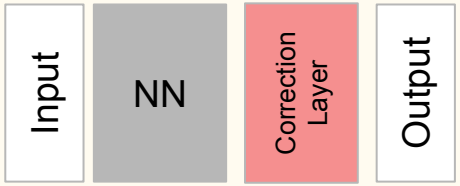
This process is called **completion**



Inequality constraints: If $g(x,y)<0$ then set y such that $g(x, y) \geqslant 0$

Example: Constraint is that $y \geq 0$. If $y_i < 0$ then set $y_i = 0$

This process is called **correction**



Our Constraints

Mass conservation - Equality constraints

Let $S = \{SO4, DU, OC, BC\}$ be the set of aerosol species. For every $s \in S$ let I_s be the indices of our output y corresponding to value of that species. Mass conservation is given by

$$\sum_{i \in Is} y_i = 0$$

Soft constraining: Add loss term $||\sum_{i \in Is} y_i||^2$

Hard constraining: Choose $j \in I_s$ and set $y_j := -\sum_{i \in Is \setminus \{j\}} y_i$

Our Constraints

Mass positivity - Inequality constraints

All predicted masses have to be positive. For our input x (masses at time 0) and our output y (change in mass), the constraint is

$$y+x \geq 0$$

Soft constraining: Add loss term $||\text{ReLU}(-(y+x)||^2$

Hard constraining: Add correction layer $\left\{\text{If } y_i+x_i < 0 \text{ then set } y_i = -x_i\right\}$

Results

Results - Log vs Linear

Better performance with log transformed data in log-scale

Transformation	Linear	Log-Scale
R^2	0.763	0.974
MSE	0.162	0.030

Original units/linear scaling probably more relevant for application

Results - Mass conservation

Soft constraining with mass loss term decreases mass violation of each aerosol species, but accuracy is decreased too

Hard constraining with completion procedure guarantees mass conservation, but also decreases accuracy

Model	Base NN	NN + mass loss	NN + completion
R^2	0.763	0.730	0.738
Mass Bias SO4	0.000011	0.0000086	0
Mass Bias BC	0.000038	0.000034	0
Mass Bias OC	0.000033	0.000011	0
Mass Bias DU	0.000001	0.00000028	0
Overall Mass Violation	0.00037	0.00014	0

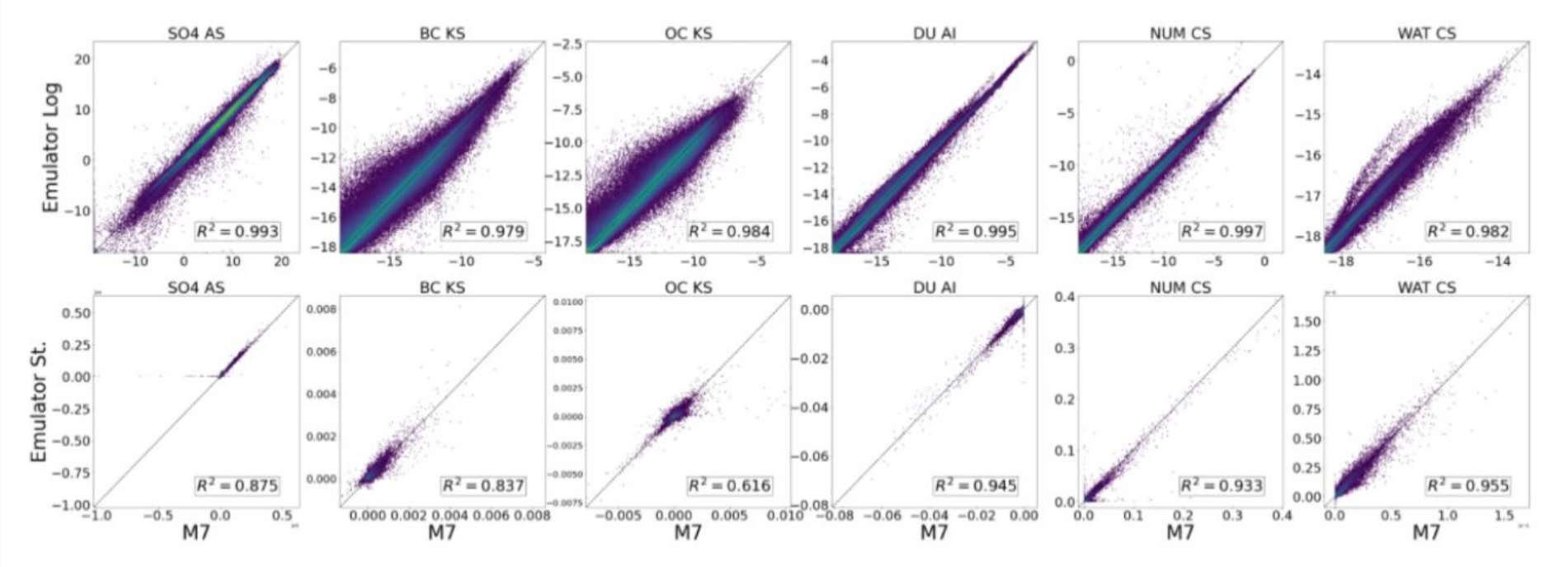
Results - Positivity

Soft constraining with positivity loss decreases negative fraction and negative mean, but also accuracy

Hard constraining with correction procedure guarantees no negative values and also increases accuracy

Model	NN Base	NN + Positivity Loss	NN + Correction
R^2	0.763	0.709	0.771
Negative Fraction	0.134	0.0894	0
Negative Mean	0.00151	0.000081	0

Results - Plots



Results - Speed-ups

Runtimes for the prediction of one global time step:

Model	M7	NN pure GPU	NN CPU-GPU-CPU	NN CPU
Time (s)	5.781	0.000517	0.0897	2.042
Speed-up	1	11181.8	64.4	2.80

Summary & Outlook

Neural networks can accurately emulate aerosol microphysics model

We can incorporate physical constraints with hard constraining to make the emulator feasible for a stable global climate model run

A significant speed-up, especially on a GPU can be achieved

Next step: Implement neural network emulator in Fortran and integrate into global climate model

Thanks for your attention!