Approximation of marine ecosystem models using artificial neural networks

Markus Pfeil, Thomas Slawig, Jaroslaw Piwonski, Maria-Theresia Verwega, Tamila Ghanbari-Ghazvini

Department of Computer Science
Christian-Albrechts-Universität zu Kiel
Future Ocean

Berlin, Feb. 2019
Motivation: Carbon cycle, $CO_2$ emissions and uptake

Figure: Simplified schematic of the global carbon cycle
Model equations for marine ecosystems

- System of transport equations for biogeochemical tracers:

\[
\frac{\partial y_i}{\partial t} = \nabla \cdot (\kappa \nabla y_i) - \nabla \cdot (vy_i) + q_i (y, u, b, d), \quad i = 1, \ldots, n_y
\]

- Solution is a steady annual cycle: \( y(t + 1) = y(t) \)

N model\(^1\)

- Only nutrients (N)
- 5 input parameter

\(^1\)[Bacastow and Maier-Reimer, 1990]
Simulation using software Metos3D³

- Single time-step:
  \[ y_{j+1} = L_{\text{imp},j} (L_{\text{exp},j} y_j + \Delta t_j q(y_j, u)) \]

- Offline simulation using transport matrices²

- One year:
  \[ y_{0+1}^\ell := y_{2880}^\ell \]

- Spin-up to reach steady annual cycle:
  - Stopping criterion \( \|y^\ell - y^{\ell-1}\|_2 < \varepsilon \)
  - or \( \ell = \ell_{\text{max}} \)

- Aim: Optimization of model parameters

- Weighted least squares cost function ...

- ... or Hellinger distance for KDEs

---
²Khatiwala et al., 2005
³Piwonski and Slawig, 2016, https://metos3d.github.io/
Application of ANNs

- ANNs: Artificial Neural Networks
- Many other terms: Machine learning, deep learning, artificial intelligence ...
- some kind of “hype” ...
- successfully used in picture/speech recognition
- autonomous driving, classification ...
- Comp. Sci. students like it ...
- 1st idea: replace optimization by ANN, student work, not successful
- 2nd idea: start with forward simulation, student work, very successful: $\mu s$ cputime instead of minutes or hours, but ...
Starting from the beginning: Model of a neuron

\[
\begin{align*}
    z &= \sum_{i=1}^{n} w_i x_i + b \\
    y &= f(z)
\end{align*}
\]

**Figure:** Model of a neuron.

**Activation function**

- **Identity:** \( f : \mathbb{R} \rightarrow \mathbb{R}, \ x \mapsto x \)
- **Exponential linear unit:**
  \[
  f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}, \ (x, \alpha) \mapsto \begin{cases} 
    x, & \text{if } x \geq 0 \\
    \alpha \cdot (e^x - 1), & \text{if } x < 0
  \end{cases}
  \]
Artificial neural network - Structure

Figure: Structure of an artificial neural network
Artificial neural network - Training

- Optimization problem: Adapt network parameters (weights between neurons)
- Aim: Reduce misfit between network output and data ("Empirical Risk") for training data set
- Classical least-squares problem
- Used Software: Framework TensorFlow in python
- Standard Optimization method for ANNs: SDG – Stochastic Gradient Descent
- Main question is the design of the network (# layers, # neurons per layer)
Artificial neural network for the N model

Figure: Fully connected network (FCN)

- Prime decomposition: $52750 = 2 \cdot 5 \cdot 5 \cdot 5 \cdot 211 = 10 \cdot 25 \cdot 211$
- First result (BSc thesis): error of 0.01 %, but: small parameter range
- Now: systematic test: 33 data sets for training data, 68 for validation (latin hypercube)
**ANN prediction of the $PO_4$ concentration**

Figure: $PO_4$ concentration, original simulation (Metos3D). Units are mmol P m$^{-3}$. 

(b) Slice plane of the Atlantic

Figure: ANN prediction
Difference ANN prediction – original simulation

(a) Surface layer (0 m to 50 m)

(b) Slice plane of the Atlantic

Figure: $\frac{\|Y_{\text{ANN}} - Y_0^{10000}\|_2}{\|Y_0^{10000}\|_2}$

Figure: $\frac{\|Y_{\text{ANN}}^{1000} - Y_0^{10000}\|_2}{\|Y_0^{10000}\|_2}$
Mass correction is necessary

Prediction of the ANN

- Containing only 76.7% of initial mass (average)
- Relative error of 0.282

Adjusted prediction

- Adjustment of the predicted concentration to conserve the overall mass:
  \[
  \bar{c}_{\text{ann}} = \frac{m_{\text{overall}}}{m_{\text{ann}}} c_{\text{ann}}
  \]

- Reduced the relative error of 0.027

Figure: \[
\frac{\|y_{\text{ANN}}^{1000} - y_0^{10000}\|_2}{\|y_0^{10000}\|_2}
\]
Reducing the error with mass adjustment of the prediction

Figure: \[ \frac{\|y_{ANN} - y_0^{10000}\|_2}{\|y_0^{10000}\|_2} \]
Using the adjusted prediction as initial value

- Adjusted prediction as initial value for spin-up
- Calculation of a spin-up with spin-up tolerance $10^{-4}$
- Reduced simulation time

Figure: Norm of spin-up with tolerance $10^{-4}$ for different initial values.
Analysis of spin-up initialized with mass-corrected ANN prediction

Figure: $\|y_{ANN}^1 - y_{ANN}\|_2$

Figure: $\frac{\|y_{ANN}^1 - y_{ANN}\|_2}{\|y_{ANN}\|_2}$
Reduced simulation time

**Figure:** Required model years of the spin-up using constant as well as adjusted ANN predictions as initial values for all parameter sets.

**Figure:** Ratio of required model years of spin-up between constant and adjusted ANN prediction as initial values for all parameter sets.
Comparison of different FCNs

Figure: Violinplot of the required model years for different FCNs.

<table>
<thead>
<tr>
<th>FCN</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>87.0%</td>
<td>82.5%</td>
<td>87.99%</td>
</tr>
</tbody>
</table>

Table: Mean values of the ratio of the required model years between constant and adjusted ANN prediction as initial value.
Comparison of different convolutional neural networks (CNN)

Figure: Violinplot of the required model years for different CNNs.

<table>
<thead>
<tr>
<th>Initial value</th>
<th>CNN 1</th>
<th>CNN 2</th>
<th>CNN 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>87.54%</td>
<td>84.73%</td>
<td>86.3%</td>
</tr>
</tbody>
</table>

Table: Mean values of the ratio of the required model years between constant and adjusted ANN prediction as initial value.
Summary and Perspectives

- Adjustment of the prediction to conserve mass
- Using the ANN to generate initial values
- Reduction of simulation time
- Influence of ANN’s structure on this reduction

Future work:
- Redesign and improve ANN structure
  Unfortunately: trial and error

Perspectives in optimization:
- Reinforced Learning: Ongoing training during optimization
- ANN output as coarse model for surrogate-based optimization