Exploring an Evolutionary Approach for Task Generation in Meta-Learning with Neural Processes

Author: Kerem Yoner (k.yoner@student.tudelft.nl)
Responsible Professor: Matthijs Spaan
Supervisor: Joery Vries

1. Research Question

How can evolutionarily-curated curriculum enhance the generalization and learning performance of neural processes in meta-learning?

2. Background

Neural Processes (NPs) combine neural networks and Gaussian processes to quickly adapt to new tasks with minimal data. They consist of an encoder that transforms context points into latent representations, a latent variable model that infers distributions over these representations, and a decoder that predicts new data points.

Meta-learning focuses on creating models that can learn new tasks rapidly from few examples by leveraging experience from related tasks. This capability is essential in scenarios where collecting extensive data is difficult or costly.

3. Problem Description

Meta-learning problem that is used to apply the evolutionary curriculum approach is the 1-D regression with Fourier functions.

$$y(x) = a_0 + \sum_{i=1}^{n} a_i \cos\left(2\pi i \frac{x - \phi_i}{T}\right)$$

Fourier functions are parameterized by the number of cosine waves, amplitudes of the waves, phases of the waves and the period.

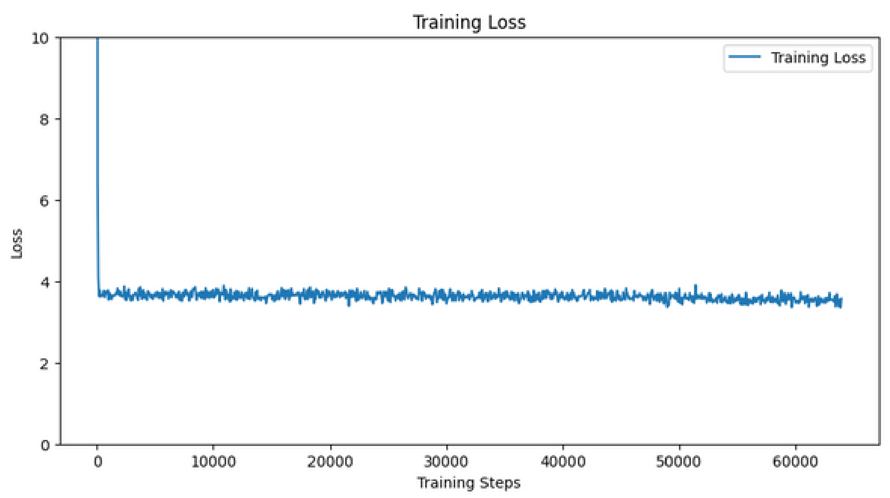
Parameter ranges for the Fourier functions determine the main task distribution that the model is learning from.

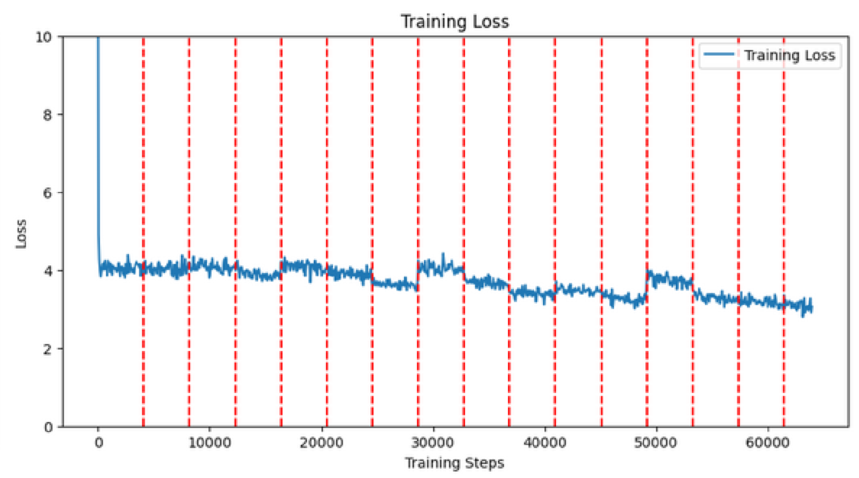
3. Methodology

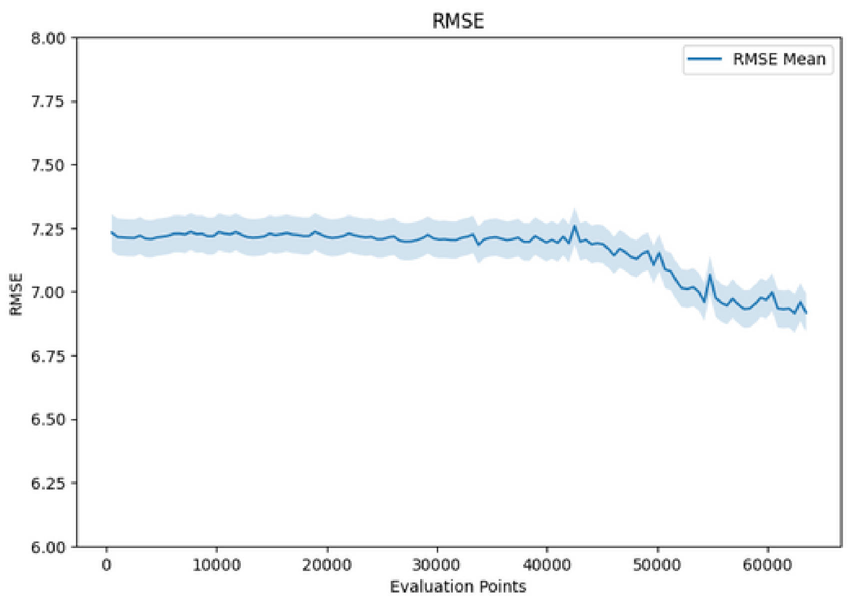
To improve Neural Processes (NPs) with an evolutionary curriculum, we followed these steps:

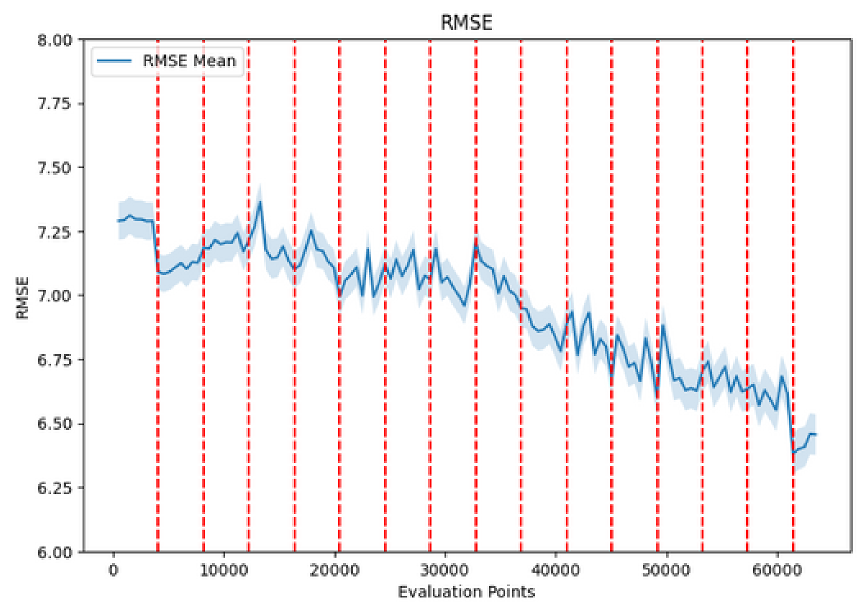
1. **Task Representation**: Define tasks using Fourier functions with parameters for the number of waves, amplitudes, phase shifts, and period. Model trains on context-target points sampled from the Fourier function that the task defines.
2. **Training Loop**: Train the NP model by introducing a new set of tasks combined by a mix of randomly sampled tasks and those generated by the evolutionary algorithm.
3. **Evolutionary Algorithm**:
   - Evaluate the fitness of tasks based on the model's performance on the task.
   - Select the top-performing candidates.
   - Apply crossover and mutation to generate new tasks.
   - Introduce the new set of tasks into the training process at regular intervals.
4. **Evaluation**: Measure model performance on new set of tasks to compare it with a baseline that is trained on randomly sampled tasks from the main distribution.

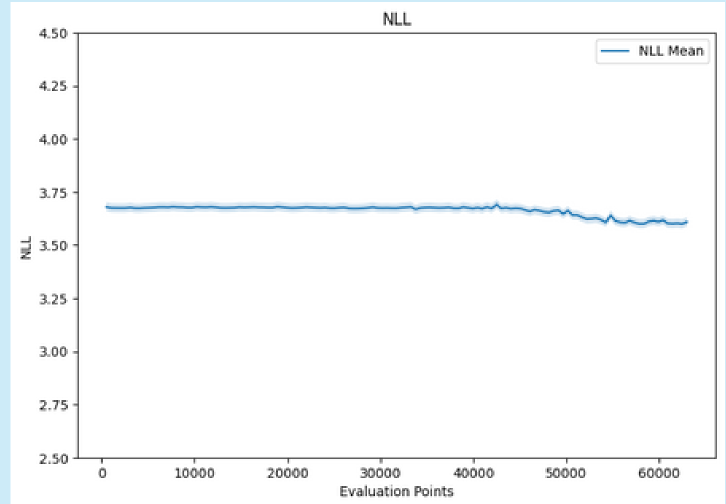
5. Results

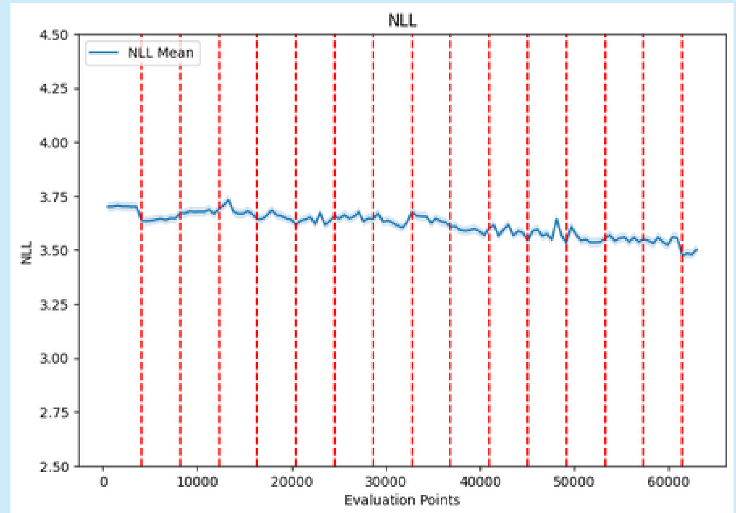












6. Conclusion

- **Training Stability**: The introduction of an evolutionary curriculum causes some instability in training, as indicated by performance spikes when new tasks are introduced.
- **RMSE Improvement**: Despite the instability, the model achieves a lower RMSE, indicating improved accuracy.
- **NLL Consistency**: The Negative Log-Likelihood (NLL) does not show significant improvement, suggesting that while predictions become more accurate, their confidence levels remain similar.

7. Future Work

- Applying the evolutionary curriculum to more complex, high-dimensional tasks.
- Investigating advanced evolutionary strategies like multi-objective optimization.
- Combining evolutionary algorithms with other meta-learning techniques.