
MACHINE LEARNING PROJECTS

- **Top 2% solution in the Featured Simulation Kaggle Competition "Lux AI"**: Ranked 19th out of 1186 teams as a solo participant ([team shmyak](#)). The solution is based on imitation and distributed reinforcement learning. It consists of:
 1. **Environment**. An OpenAI gym wrapper for a Kaggle environment, which does preprocessing of raw data to provide ready observations to a trainer agent and contains several rule based agents. Several workers use the environment to collect experience to a data buffer.
 2. **Trainer**. It performs training of a function approximator, it includes different implementations of actor-critic and policy gradient algorithms, a custom training loop for imitation learning, and a custom buffer to store game experience. The buffer uses tfrecords files to prevent storing all experience trajectories in memory but consuming them efficiently from a storage device. It uses EfficientNetV2 squeeze-and-excitation layers as a function approximator.
- **Developed a policy gradient based reinforcement learning algorithm for a custom environment**: The implementation is close to [IMPALA](#). It includes a custom [training loop](#) of an off-policy actor-critic algorithm with n-step update, policy gradients correction, entropy, and other improvements. It uses different convolutional neural nets as a function approximator and applies multi-attention for data preprocessing.
- **Developed a DQN based reinforcement learning algorithms**: The project includes custom training loops [implementation](#) of several reinforcement algorithms (TensorFlow): DQN, FixedDQN, DoubleDQN, DoubleDuelingDQN, categorical DQN. It uses [RAY](#) to distribute calculations and [DM Reverb](#) as a data buffer to perform Prioritized Experience Replay. It uses a [sparse MLP](#) as a function approximator.
- **Conducted an optimization study**: The research involved developing a model that predicts the ocean's ability to absorb carbon dioxide. It uses a custom function approximator and a non-linear least-squares minimization.
- **Conducted a signal processing study**: It proposes the method to predict 'Freak waves' based on waves parameters. The study uses cluster analysis to categorize waves to groups and then uses Fourier and Wavelet analysis to study properties and features of these groups.

RESEARCH, DEVELOPMENT AND DATA ANALYSIS EXPERIENCE

- **Helmholtz-Zentrum in Geesthacht** Geesthacht, Germany
PhD student 2017 - 2020
 - **Research**: Studied biogeochemical interactions between the ocean and the atmosphere in the North Sea region. Published 4 peer-reviewed articles, 2 preprints, and 2 jupyter notebook studies ([first](#), [second](#))).
 - **Development**: Built and optimized ocean ecosystem and biogeochemical models.
 - **Data analysis**: Processed and analyzed geospatial data using Pandas, Numpy, Matplotlib, etc. Performed Exploratory data analysis in Jupyter notebooks.
- **Institute of Oceanology** Moscow, Russia
Research engineer 2014 - 2017
 - **Research**: Studied waves, Arctic ecosystems and biogeochemical processes. Published 5 peer-reviewed articles.
 - **Development**: Introduced the CMake build system and refactored Fortran90 code to modern object-oriented Fortran in multiple numerical projects. Introduced dynamic NetCDF i/o into several Fortran projects. Developed a diffusion-advection model of solutes and particulates transport in the ocean.
 - **Data analysis**: Processed, analyzed and visualized waves data in MatLab using Signal Processing Toolbox and Wavelet Toolbox.

PROGRAMMING SKILLS

- **Languages**: Python (Tensorflow, Keras, Numpy, Pandas), FORTRAN, MatLab, LaTeX

PUBLICATIONS

- [A google scholar profile](#).

EDUCATION

- **Moscow State University** Moscow, Russia
Specialist, Oceanography 2003 - 2008
- Online courses: [Deep Learning specialization](#), [Machine Learning](#), [Bayesian Statistics](#).