MOISEY ALAEV

UCLA September 2018 - June 2022
B.S. Mathematics of Computation and Minor in Statistics
GPA: 3.5

Los Angeles, CA
917-355-7202
moiseyalaev@gmail.com
moiseyalaev.com
github.com/moiseyalaev
inkedin.com/in/moiseyalaev

SKILLS

- **Software Engineering:**
  - C, C++
  - Python, Java
  - Git, Bash
- **AI Dev:**
  - TensorFlow Developer Certified
  - PyTorch, Keras
  - Scikit-Learn
  - OpenCV
  - NumPy, SciPy
  - R, Pandas
  - Basic SQL and pyTaco
  - Anaconda, PyCharm
- **Web Dev:**
  - HTML/CSS:
    - EJS/Handlebars
    - Bootstrap
  - Javascript:
    - Node.js
    - Express.js
    - MongoDB
- **LaTeX**
- **Fluency in Russian**

COURSEWORK

- **Computer Science:**
  - Data Structures
  - Software Construction Lab
  - Artificial Intelligence
  - Algorithms and Complexity
- **Mathematics:**
  - Discrete Structures
  - Graph Theory
  - Machine Learning
  - Data Theory
- **Statistics:**
  - Probability
  - Data Analysis and Regression
  - Statistical Models and Data Mining

PROFESSIONAL EXPERIENCE

UCLA MACHINE LEARNING LAB

RESEARCH ASSISTANT JANUARY 2020 - PRESENT

- Researching Semi-Supervised models for Nonnegative CP Decomposition of Tensors (SSNCPD: a form of tensor factorization) using *TensorFlow*, *PyTorch*, and *NumPy* backends.
- Co-leading a software team on developing an open-source Python package of multiplicative update methods for SSNCPD. Expected to be released at the end of 2021.
- Testing package on echo-cardiogram videos that were preprocessed into tensors with packages from *OpenCV* and raw data tensors from *Kaggle* which are imported using *pyTaco*.

STEAM:CODERS

INTERN & TUTOR OCTOBER 2020 - JULY 2021

- Initiated and developed a data system using *MongoDB* of student information (attendance, quiz scores, etc.) from *JSON files* so faculty can utilize student data to improve future instruction.
- Spearheaded and implemented a new curriculum in machine learning, exceeding enrollment expectations by a magnitude of three due to high student engagement.
- Utilized *Scratch*, *Keras (Python)*, *Javascript*, and *Java* for instruction.

ACM-AI

OFFICER | SEPTEMBER 2019 - 2020; MEMBER | SEPTEMBER 2018 - PRESENT

- Organized the Advanced AI learning track: taught UCLA students difficult ML concepts: *activation and loss functions*, *backpropagation*, *CNNs*, *RNNs*, *NLP*, *transfer learning*, etc.
- Engaged in advanced AI workshops from freshman year until promotion to an officer position.
- Currently partaking in the "Apply ML" track, learning new tools and ways to develop ML models.

PROJECTS

CNN RESEARCH AND COMPUTER VISION EXPERIMENTS 2020 - 2021

- Researched the mathematical foundations of Convolutional Neural Networks (CNNs) such as *convolution*, *pooling*, *flattening*, *dropout*, and *fully connected NN layers*.
- Conducted initial experiment with a *CNN* using *Keras* to train the handwritten *MNIST dataset*.
- Integrated code, math derivations, and graphs cohesively into a research paper using *LaTeX*.
- Extended the project over the course of 2021 to several *computer vision* problems that employed *transfer learning*, *ImageDataGen*, and *Image Augmentation* with *TensorFlow*.

TIME SERIES FOR PREDICTIONS 2021

- Implemented two time series models with *RNNs* using *Tensorflow* in *PyCharm IDE*.
- First: Predicted Google's Stock Price with *LSTM* and *Dropout* layers.
- Second: Predicted Sunspots with *multi-bidirectional LSTMs*, *Convolutions*, and *Lambda* layers.
- Preprocessing data with *Pandas* and *Sklearn;* creating *windowed datasets;* plotting series data.

AUTONOMOUS CAR 2020

- Developed a *neural network* model using *PyTorch* on *Anaconda's Spyder IDE* that implements *Reinforcement Learning* concepts such as *Q-learning*, *Memory Replay*, and *Markov Processes*.
- Adapted the model to a *GUI* of the mini-game: As the car drives, the user can add "sand" to indicate where the car should avoid driving. Over time, the car learns to avoid the sand to get to its destination faster.

ZOMBIE-DASH GAME 2019

- Designed a large-scaled *C++* game implementing several software development concepts such as *C++'s STL* (for *data structures*), *polymorphism and inheritance, pointer operations, GUI's, and algorithmic complexity*.