Benyamin Gidanian

Los Angeles, CA 90034 (310) 613 3655 bgidanian@gmail.com
GitHub: github.com/ben-gid
LinkedIn: linkedin.com/in/benyamin-gidanian-46305a364/

Summary

Analytical Python & ML Developer dedicated to producing efficient, reproducible, and verifiable code. Grounded in a rigorous CS foundation from Harvard (CS50) and Stanford (Machine Learning Specialization), focusing on the “how and why” of algorithmic behavior to ensure system reliability. Experienced in developing high-performance ML models using TensorFlow, Scikit-learn, and XGBoost and in building traditional full-stack applications with Django and Flutter. Currently focused on bridging the gap between theoretical AI and functional software by creating modular, integration-ready models designed for production environments.

Technical Skills

Languages: Python, Dart(flutter), JavaScript, SQL, C, HTML5, CSS3
Machine Learning: Supervised Learning (Regression/Classification), Unsupervised Learning, Recommender Systems, Reinforcement Learning, Neural Networks, Decision Trees, XGBoost
Deep Learning / NLP: PyTorch, Torchvision, TensorFlow, Keras, scikit-learn, BERT, Attention Mechanisms, NLTK, Tokenization
Data Engineering: NumPy, Pandas, Data Preprocessing, Feature Engineering, Evaluation Metrics (Precision/Recall, F1-Score)
Web Development: Django, FastAPI,Flask, React, DOM Manipulation, Database Design
Developer Tools: Git, GitHub, Linux, Bash, Jupyter Notebooks, Debugging, Testing

---

TECHNICAL PROJECTS

ML Fundamentals Library | Python, NumPy, Matplotlib, Pytest GitHub Link

- **Architected** a modular machine learning library from scratch using **NumPy**, implementing 6+ core algorithms (Linear/Logistic Regression, K-Means, Decision Trees, Neural Networks) without high-level frameworks.

- **Developed** a custom Deep Learning engine featuring manual backpropagation and a Keras-inspired Sequential API for model building.

- **Optimized** mathematical operations through vectorization, significantly reducing computational overhead compared to iterative implementations.

- **Ensured** production-grade code quality by maintaining a 100% pass rate on unit tests (Pytest) and building a custom AST-based documentation generator.

Deep Q-Network (DQN) for Autonomous Lunar Landing | Reinforcement Learning, TensorFlow, Gymnasium, Python GitHub Link

- **Implemented** a DQN Agent from scratch using **TensorFlow/Keras** to solve the Gymnasium LunarLander-v3 environment, achieving a "solved" status with a rolling average score of 200+.

- **Engineered** stability mechanisms including **Experience Replay** (using collections.deque) to break temporal correlations and **Target Networks** with Soft Updates (τ=0.001) to prevent model oscillations during training.

- **Optimized** training performance by utilizing **tf.function** decorators for graph-mode execution, significantly reducing per-episode training time.

Full-Stack Social Networking Platform (Django & React) GitHub Link

- **Designed** and implemented a scalable relational database schema using **Django ORM**, managing complex many-to-many relationships for user follows and post interactions.

- **Developed** a RESTful API used by a **React** frontend to handle asynchronous data updates (likes, edits) without page reloads, improving UX latency.

- **Engineered** server-side pagination and optimized database queries using **select_related** and **prefetch_related** to ensure performant data retrieval.

- **Implemented** secure user authentication and session management using **Django's** security framework.

Crossword Puzzle Generator (AI/Search Optimization) | AI/Search Optimization, Python GitHub Link

- **Implemented** a Constraint Satisfaction Problem (CSP) solver to automatically generate complex crossword puzzles using Python.

- **Engineered** a backtracking search algorithm optimized with AC-3 (Arc Consistency) to reduce search space by over 80%.

- **Integrated** Heuristics including Minimum Remaining Values (MRV) and Least Constraining Values (LCV) to enhance solver efficiency and handle high-density grids.

PROFESSIONAL EXPERIENCE

Talmud Tutor (One-on-One Mentor) | Mesivta Bircas Yitzchok | Sept 2023 – March 2025

- Analyzed and simplified intricate logical frameworks and multi-layered arguments.
- Applied systematic debugging of theoretical scenarios to identify edge cases and logical fallacies.
- Developed personalized curricula to optimize learning speed and conceptual retention.
- Maintained a 1.5-year track record of reliability and clear communication in high-pressure environments.

EDUCATION & CERTIFICATIONS

Machine Learning Specialization | Stanford University & DeepLearning.AI (Coursera)

- Supervised Learning, Neural Networks, Decision Trees, Unsupervised Learning, Recommender Systems.

CS50 Computer Science Suite | Harvard University (EdX) * **CS50x:** Introduction to Computer Science (C, Python, SQL, Data Structures, Algorithms).

- **CS50p:** Programming with Python (Unit Testing, OOP, Regular Expressions).
- **CS50ai:** Introduction to Artificial Intelligence (Search, Logic, Probability, Optimization).
- **CS50w:** Web Programming with Python and JavaScript (Django, React, CI/CD).
