Tanekella Anjali

tanikellanjali@gmail.com| <https://www.linkedin.com/in/tanekellaanjali/> | <https://medium.com/@anjalitanikella> | https://github.com/tanikellanjali

Aspiring PhD candidate with a strong foundation in AI, machine learning, and big data systems. Experienced in developing AI-driven projects, NLP systems, and reinforcement learning models. Passionate about prediction algorithms, distributed AI systems, and optimizing large language models (LLMs) for real-world applications. Interested in advancing research in fractal-based AI, GPU optimization, and reinforcement learning for LLM fine-tuning.

**INTERESTS:**

* AI and Machine Learning Optimization
* Reinforcement Learning and Fine-Tuning
* Large Language Models (LLMs)
* Pattern Recognition and Prediction Algorithms
* Big Data and Distributed AI Systems
* NLP and Generative AI

**SKILLS:**

* Programming: Python, SQL, R, NoSQL, MySQL, Javascript
* Machine Learning/AI: TensorFlow, PyTorch, Hugging Face, scikit-learn, Reinforcement Learning, NLP (spaCy, NLTK)
* Big Data Technologies: Apache Spark, GCP (BigQuery, Dataflow, Pub/Sub), AWS
* Visualization: Tableau, Power BI, Looker, Matplotlib, Seaborn
* Fullstack: React JS , Node JS , Express JS , Flask
* ETL and Pipelines: Airflow, GitLab, Cron, Data Warehousing
* Database: PostgreSQL, PGAdmin, DBeaver, Oracle, MySQL Workbench
* IDE: Jupyter Notebook, VS Code, Google Collab

**ARTICLES :**

* **Harnessing Fractal Geometry in AI: The Future of Efficient Computing and Problem Solving**

**Dec 20 2024**

**Overview:**

This article explores the application of fractal geometry in AI to enhance computational efficiency, optimize problem-solving algorithms, and improve memory utilization. It discusses the role of logarithmic spirals and tree fractals in revolutionizing machine learning and robotics.

**Key Concepts:**

Fractals in AI: Recursive structures for efficient computing and pattern recognition.

Logarithmic Spirals: Inspire dynamic algorithms and optimization techniques.

Tree Fractals: Improve hierarchical problem-solving and decision models.

Applications: Fractal neural networks enhance pattern recognition, predictive modeling, and robotics.

**Tags:** AI Optimization, Fractal Geometry, Machine Learning, Neural Networks, Robotics

* **Apple’s AI Hardware Revolution: On-Device Intelligence Without Centralized Data Storage**

**Dec 20 2024**

**Overview:**

Apple’s AI strategy focuses on on-device intelligence, eliminating the need for centralized data storage to prioritize privacy and efficiency. The article highlights Apple's silicon advancements and federated learning techniques.

**Key Concepts:**

Apple Silicon (M1, M2): Optimized for AI workloads, with UMA and Neural Engine.

Core ML and Metal: Accelerates model training and inference.

Privacy-Focused AI: On-device fine-tuning, federated learning, and secure enclaves.

Applications: Health apps, smart assistants, AR tools, and AI personalization.

**Tags:** On-Device AI, Apple Silicon, Privacy, Federated Learning, Core ML

* **The AI Chip Wars: NVIDIA, AMD, and Apple’s Innovations in LLM Fine-Tuning**

**Dec 20 2024**

**Overview:**

This article compares the AI hardware race among NVIDIA, AMD, and Apple, focusing on LLM fine-tuning, chip performance, and market positioning.

**Key Concepts:**

NVIDIA: Dominates AI compute with Blackwell GPUs, CUDA frameworks, and DGX systems.

AMD: Rising with MI325X AI accelerators and ROCm platform for cost-effective AI.

Apple: Prioritizes on-device AI with silicon optimized for privacy and efficiency.

Applications: Enterprise AI, generative models, autonomous systems.

**Tags:** AI Hardware, LLM Fine-Tuning, NVIDIA, AMD, Apple, GPUs

* **Exploring Row-Wise Pattern Recognition for Stock Analysis**

**Dec 8 2024**

**Overview:**

The article details how row-wise pattern recognition can identify trends, anomalies, and future stock movements by analyzing sequential data.

**Key Concepts:**

Pattern Extraction and Filtering: Identifies trends by extracting subsequences.

Applications: Predictive modeling, volatility analysis, and sector clustering.

Challenges: Data noise, dynamic markets, and large dataset processing.

**Tags:** Stock Analysis, Pattern Recognition, Sequential Data, Predictive Modeling

* **RAG vs RLFT: Which Is Better for AI Models, and What Processors Are Needed?**

**Oct 11 2024**

**Overview:**

A comparison of Retrieval-Augmented Generation (RAG) and Reinforcement Learning Fine-Tuning (RLFT) for LLM optimization, detailing their use cases and hardware requirements.

**Key Concepts:**

RAG: Best for fact-based tasks requiring external knowledge retrieval.

RLFT: Ideal for adaptive, personalized AI systems learning from user feedback.

Processors: NVIDIA A100, RTX 3090, and multi-GPU setups.

**Tags:** RAG, RLFT, Large Language Models, Fine-Tuning, AI Processors

* **Understanding the Math Behind LLM Models and Fine-Tuning**

**Oct 11 2024**

**Overview:**

This article breaks down the mathematical foundations of large language models (LLMs), focusing on transformers, attention mechanisms, and optimization techniques used in pre-training and fine-tuning.

**Key Concepts:**

Self-Attention: Key to capturing long-range dependencies.

Positional Encoding: Ensures sequential understanding.

Fine-Tuning with PPO: Reinforcement learning fine-tuning for model improvement.

**Tags:** LLM, Transformers, Fine-Tuning, PPO, Neural Networks

* **Why GPUs are Essential for Training Large Language Models: Cost, Performance, and Efficiency**

**Oct 11 2024**

**Overview:**

This article explores the importance of GPU size and architecture in training large language models (LLMs) like GPT, T5, and BERT. It shares real-world experiences of GPU memory issues and highlights which GPUs are best suited for different model sizes and AI applications.

**Key Concepts:**

GPU Memory Matters: Critical for handling large models and batch sizes.

Performance Bottlenecks: Insufficient GPU memory can lead to CUDA errors and slow training.

Precision and Parallel Processing: Mixed precision and multi-GPU setups improve efficiency.

Model-Specific GPUs: RTX 3090 for small to medium models, A100 for large models, and multi-GPU for massive LLMs.

**Tags**: LLM, GPU, NVIDIA A100, Model Training, Hardware Optimization

* **Why Reinforcement Learning Beats Regular Fine-Tuning for AI Model Training**

**Oct11 2024**

**Overview:**

Reinforcement learning (RL) surpasses traditional fine-tuning by allowing models to adapt dynamically through real-time feedback, maximizing long-term performance, and optimizing responses to evolving environments.

**Key Concepts:**

Dynamic Adaptation: RL enables models to learn from continuous feedback rather than static datasets.

Long-Term Optimization: RL focuses on cumulative rewards, unlike fine-tuning which targets immediate results.

Exploration and Discovery: RL encourages models to explore diverse solutions.

Continuous Improvement: Models evolve in real-time through ongoing interactions and feedback.

**Tags:** Reinforcement Learning, Fine-Tuning, PPO, AI Optimization, Dynamic Training

* **Reinforcement Learning Fine-Tuning: The Future of Adapting Language Models**

**Oct 11 2024**

**Overview:**

This article explains how RLFT (Reinforcement Learning Fine-Tuning) refines LLMs like T5 using real-time feedback. A step-by-step guide on fine-tuning with Proximal Policy Optimization (PPO) is provided, demonstrating the benefits of dynamic learning and user-driven model improvement.

**Key Concepts:**

RLFT Benefits: Real-time feedback, task-specific adaptation, and dynamic learning.

Gradient Accumulation: Addresses small dataset constraints by simulating larger batch sizes.

Code Example: Demonstrates PPO fine-tuning for T5 models.

Continuous Feedback Loop: Models evolve through iterative interactions.

**Tags:** RLFT, PPO, LLM Fine-Tuning, Feedback Loop, Adaptive AI

**PROJECTS :**

* **JARVIS – Chatbot**

**Status: Delivered on Dev 6 20204**

**Client: Verizon**

**Overview:**

JARVIS is a web-based tool designed to enable users to interact with multiple databases using natural language queries related to VSAD. The front end is built with React.js, while the backend utilizes Node.js and Express.js. PostgreSQL serves as the primary database, with an additional Flask NLP service for query processing. The application employs prompt engineering and fine-tunes LLM models for enhanced query interpretation. A metadata table containing schema, table names, and column details was created for model training.

**Tech Stack:** React.js, Node.js, Express.js, PostgreSQL, Flask, LLM (BERT-like), Fine-tuning, Prompt Engineering

**Future Enhancements:**

Expand LLM model coverage across additional databases.

Implement real-time query optimization.

Enhance multi-agent approach for simultaneous query handling.

* **Pattern Recognition Algorithm**

**Status: Delivered July 20 2024**

**Client: Verizon**

**Overview:**

A system for analyzing time-series data to extract recurring patterns from alerts or event logs. The algorithm uses GPU acceleration through PyTorch and processes subsequences to identify patterns, which are then stored in PostgreSQL for further analysis.

**Tech Stack:** PyTorch, pandas, numpy, psycopg2, PostgreSQL

**Future Enhancements:**

Introduce anomaly detection with LSTM models.

Automate subsequence classification using Transformer models.

Scale pattern extraction for large distributed datasets.

* **JENGA - AI-based Project Management Platform**

**Status: inProgress since Nov 11 2024**

**Client: Verizon**

**Overview:**

An interactive project management tool inspired by Jenga, where management can post project ideas, and teams bid to take on tasks. The platform offers project visualization, resource planning, and competitive bidding. Projects are represented as "bricks" in a virtual stack, enhancing engagement and transparency.

**Tech Stack:** React.js, Node.js, PostgreSQL, Flask, AI-driven resource allocation models

**Future Enhancements:**

Integrate AI-driven project matching.

Add predictive analytics for project success rates.

Implement NLP summarization for project briefs.

* **Loan Prediction System**

**Client: University at Buffalo**

**Overview:**

A system to predict loan approvals using Random Forest classifiers. The project preprocesses loan application datasets, encodes categorical variables, and scales numerical features. The model identifies patterns in applicant income, education, and other features to make predictions.

**Tech Stack:** pandas, sklearn, Random Forest, numpy

**Future Enhancements:**

Implement Gradient Boosting models.

Introduce deep learning models (TabNet) for enhanced accuracy.

Incorporate real-time feedback to improve predictions.

* **WARMACHINE - Code Prediction and Auto-completion**

**Status: POC delivered on Oct 28 2024**

**Client: Verizon**

**Overview:**

A code prediction system that fine-tunes LLM models from Hugging Face using reinforcement learning (RL) techniques like PPO. The model assists with code documentation, debugging, and feature requests by generalizing across diverse datasets. A feedback loop rewards correct code generation and penalizes errors, enhancing auto-completion.

**Tech Stack:** Hugging Face (LLM), PyTorch, PPO, Flask, Node.js

**Future Enhancements:**

Expand dataset to cover broader programming languages.

Integrate automated unit testing during model feedback loops.

Develop VS Code and JetBrains plugins for real-time suggestions.

* **Report Automation**

**Status: Delivered on Nov 15 2024**

**Client: Verizon**

**Overview:**

Automates downloading, processing, and inserting CSV files from Tableau dashboards into PostgreSQL. The script handles directory movement, data cleansing, and schema creation, ensuring efficient reporting.

**Tech Stack:** Python, pandas, psycopg2, Tableau, PostgreSQL

**Future Enhancements:**

Implement AI-driven anomaly detection in reports.

Automate data visualization using Plotly or Dash.

* **AI Collaboration Social Media Platform**

**Status: In progress since Nov 12 2024**

**Client: Personal Project**

**Overview:**

A social media platform designed for AI specialists to collaborate, post, and solve problems. It integrates SSO, role-based access, and community-driven posting features. Users can share resources, participate in threads, and receive managerial approval for posts.

**Tech Stack:** React.js, Flask, PostgreSQL, WebSockets, NLP (Auto-tagging)

**Future Enhancements:**

Implement ML for auto-tagging posts.

Develop mobile applications for Android/iOS.

Add gamification with badges and leaderboards.

* **Client Job Referral System**

**Status: POC delivered on**

**Client: Infinite Computer Solutions**

**Overview:**

A web app allowing clients to post job openings and automate email blasts to recipients for referrals. The platform includes job submission, automated email handling, and referral form integration, ensuring seamless candidate referrals.

**Tech Stack:** React.js, Node.js, Nodemailer, PostgreSQL

**Future Enhancements:**

Implement AI-based resume screening.

Add job-matching algorithms for candidate recommendations.

Expand referral tracking with performance analytics.

**RELEVANT WORK EXPERIENCE :**

**ML engineer / Python Developer |Infinite Computer Solutions| Basking Ridge , NJ | March 2024**

**Client : Verizon**

Tech Used : Airflow , Pgadmin , DBeaver , Gitlab , juypeter python , ML , genAI , prompt engineering , API , SQL, Cron , Pandas, NumPy, TensorFlow, PyTorch, Scikit-learn, MapReduce, Spark, Plotly, Hugging Face, Llama, NLP, Looker, Grafana, Excel, React JS , Node JS , ExpressJS , Flask JS , Java script

* Led development of AI-powered JARVIS chatbot integrating NLP and LLM fine-tuning to streamline database queries and alert management, reducing mean time to resolution (MTTR) by 15%.
* Engineered WARMACHINE, a code prediction system utilizing reinforcement learning (PPO) for auto-completion and debugging, improving developer productivity by 20%.
* Designed and deployed an AI-driven project management platform (JENGA), incorporating predictive analytics for resource allocation and bidding.
* Automated reporting workflows by developing ETL pipelines that process Tableau CSV exports and ingest data into PostgreSQL, resulting in 30% faster reporting cycles.
* Collaborated with a cross-functional team to introduce and leverage Meta Llama 3 for developing AI-driven solutions within the organization.
* Engineered and implemented generative AI-based projects including alert summarization tools and interactive chatbots to streamline the Software Development Life Cycle (SDLC) process. Reduced MTTR (mean time to resolution )by 15 %
* Conducted in-depth data analysis and employed machine learning models to identify recurring alert patterns and dependencies across multiple applications.
* Developed a GPU-accelerated algorithm to efficiently process large datasets and discover patterns, resulting in a 75% increase in alert handling speed and a significant reduction in manual effort by 50-90%
* Leveraged pattern analysis to proactively identify potential anomalies and outages, enabling early intervention and prevention.
* Designed and implemented an alert summarization system using Meta Llama 3 and prompt engineering to provide concise and informative summaries for production teams. Increased analyst productivity by 30% through the development of automated alert summarization
* Developed an RL-powered chatbot capable of self-learning and providing up-to-date information on data, codebase and DAGs through continuous interaction with internal data sources.

**Data Scientist| Thermofisher | Grand Island , NY | Feb 2024 – March 2024**

Tech Used :Big Query , Data warehousing , Airflow , ETL , Python , SQL , ML

* Utilized BigQuery to extract, transform, and load (ETL) over 5TB of data from various sources (databases, APIs, file storage) for data warehousing and analysis.
* Developed and maintained data pipelines for migrating, verifying, cleaning, and loading nearly 10,000 attribute data points into an Oracle Cloud data warehouse, ensuring data integrity and efficient data flow. Utilized Apache Airflow to orchestrate and schedule these pipelines for reliable data warehousing automation.
* Developed 18 machine learning models leveraging techniques like linear regression, random forest, and neural networks (
* Compared model performance using evaluation metrics to select optimal models for data warehousing, considering factors like interpretability and explain ability.
* Created data validation checks, unit tests, and monitoring dashboards covering data quality, model performance, and platform reliability.

**Database Engineer Assistant | University at Buffalo | Buffalo, USA | Nov , 2022 – April, 2023**

Tech used : BigQuery , ETL , SQL , python , oracle

* Designed and implemented over 10 relational database schemas adhering to 3NF principles, ensuring data integrity for seamless migration to BigQuery's scalable data warehouse
* Migrated 4 on-prem SQL Server databases to BigQuery, reducing latency by 40% and enabling unlimited scalability. Utilized Airflow to orchestrate and automate the data pipeline for reliable data movement.
* Built 5 production ETL pipelines with Python and BigQuery, processing over 10 TB of transaction data daily. These pipelines ensured efficient data ingestion, transformation, and loading into the cloud data warehouse.
* Wrote over 50 stored procedures and 100+ complex SQL queries, contributing to a 30% reduction in grading calculation reporting time. This experience translates well to writing and optimizing queries for BigQuery
* Improved database query performance in Oracle DBMS by 45% on average through indexing, partitioning, and optimization techniques. This proficiency in database optimization is transferable to optimizing BigQuery queries for efficient data retrieval.

**Data Scientist | DXC Technologies | Bangalore, India | June , 2020 – December , 2021**

Tech Used : BigData processing , pyspark , SQL , CI/CD , JSON , VBA , excel , Service Now , data warehousing , Dashboard building , budget reporting , Agile , resource allocation , ETL pipeline , data modeling

* Built 2 analytical datasets on 5 TB of financial data enabling 30+ business users to perform ad-hoc analysis with sub-second query speeds
* Applied machine learning on over 100 million daily IoT events to classify customer usage patterns (PySpark Random Forest models)
* Automated the generation of 15+ financial variance reports through Excel VBA macros, providing insights into budget vs. actuals
* Designed optimized JSON data integration flows linking 3 critical finance platforms, through requirements gathering with 6 stakeholder groups
* Maintained 99.99% uptime for 10+ ETL pipelines spanning source systems like ServiceNow to cloud data warehouses
* Democratized data science across the organization by building self-serve Power BI dashboards on top of BigQuery SQL analytics
* Handled CI/CD processes for data pipelines, ensuring rigorous validation before deployment to production

**Internship Trainee | TCS e-Serve International Ltd | Hyderabad Area, India | May ,2019 – Nov, 2019**

Tech Used : QUIC , JAVA , HTTPS , TCP , UDP , IP model

* Performance Analysis of QUIC Protocol: Conducted in comparison with HTTPS and HTTP servers, focusing on speed and security aspects.
* HTTPS Technology: Explored SSL integration with HTTP, emphasizing authentication, security certificates, and its reliance on TCP (Transmission Control Protocol) for reliable delivery.
* TCP and UDP Protocols: Highlighted TCP as a reliable, ordered delivery protocol in contrast to UDP (User Datagram Protocol) which does not ensure delivery order.
* QUIC Protocol Basis: Described QUIC protocol's foundation on UDP, acknowledged for its speed and potential enhanced security, as discovered by Google.
* Java Implementation: Mentioned the usage of Java for the study, indicating its role in the practical analysis of the QUIC protocol's performance.

**PUBLICATIONS:**

**Title:** Performance Analysis of QUIC Protocol in Comparison with HTTPS and HTTP Servers

**Tech Stack:**

* QUIC Protocol (Quick UDP Internet Connections)
* TLS (Transport Layer Security)
* HTTP/HTTPS
* UDP/TCP
* OSI and IP Model

**Publication:**

Journal : International Journal of Scientific and Engineering Research, volume 11 , issue 10 October 2020

Guided by Pranay Sahay

Mentored by Manikanth Koganti

Authored by T. Anjali

**Overview:**

* The research paper explores and compares the performance of Google's QUIC protocol with HTTPS and HTTP servers. The study focuses on the speed of data transfer, connection time, and data loss. QUIC, which is based on UDP, aims to simplify and accelerate connections while providing comparable or improved security over HTTPS, which relies on the TCP protocol.
* **Approach:**
* The paper begins with foundational knowledge on the OSI and IP models, as well as the fundamental differences between TCP and UDP.
* A step-by-step breakdown of HTTP, HTTPS, and TLS security protocols is provided to highlight existing communication inefficiencies.
* The core analysis focuses on how QUIC eliminates redundant handshakes by integrating TLS directly into the protocol, reducing connection times and minimizing head-of-line blocking.
* The research involved building and testing a QUIC client-server model using tools from the Chromium project. Comparative tests were conducted by measuring the speed of data transfer and packet loss between QUIC and HTTPS.

**Conclusion:**

The analysis demonstrated that QUIC significantly enhances data transfer speeds, reducing latency by 3.5-8% compared to HTTPS. Additionally, data buffering and loss were reduced by over 50%, underscoring QUIC’s efficiency in sending more messages within shorter timeframes. This paper concludes that QUIC could serve as a valuable advancement in internet communication protocols by offering faster, more secure, and resilient connections.

**VOLUNTEERING EXPERIENCE :**

**Data Engineer(Volunteering) | Community Dreams Foundation | Florida, USA | October 2023 – Present**

Tech Used : Python, SQL ,Spark, GCP (BigQuery, Dataflow, Pub/Sub) ,Pandas, NumPy, scikit-learn ,Looker, Tableau , Database , DBMS, SQL, normalization , Beautifulsoup

* Harvested over 300K rows of diverse data from 50+ sources via Python and web scraping, centralizing it into a scalable data repository on BigQuery. Utilized Airflow to orchestrate and schedule the data ingestion process, ensuring reliable data collection.
* Ingested a 5-year municipal energy dataset with usage details for advanced analytics in BigQuery. Leveraged Airflow to automate data loading, enabling timely access for analysis.
* Standardized 8 departmental datasets with pandas/NumPy, enhancing data quality for cross-functional insights before loading them into BigQuery.
* Built 5 Looker dashboards fueled by BigQuery data and statistical models, aligning KPIs with organizational goals.
* Developed forecasting models on 2 years of historical budgets, improving projection accuracy by 15%. Explored the possibility of integrating these models with BigQuery for future data analysis.
* Constructed ETL pipelines on GCP, achieving 99% uptime and seamless orchestration across business units. Utilized Airflow to schedule and manage these pipelines, ensuring efficient data flow into BigQuery
* Utilized Snowflake and MySQL Workbench for data modeling, experience transferable to BigQuery schema design

**EDUCATION:**

* Master’s in professional studies: Data Science and Application (SUNY at Buffalo, NY (May ,2023))

Coursework : Numerical methodologies , ML , SQL , data base , Probability , statistics , analysis

* Bachelors in Technology : Electronics and Communication Engineering (HITAM , Hyderabad (June , 2020))

Coursework : Applied Physcis , Calculus , Analog circuits , network analysis , digital logic design , Electromagnetic transmissions , Linear and digital IC IOT , Micro Controller , Antenna and wave propagation , VLSI , Microwave and optical communication .