Nurmukhamed Ashekey

AI Engineer

Almaty, Kazakhstan | github.com/nurmukhamedkz | linkedin.com/in/nurmukhamed-ashekey-3031a3369

PROFESSIONAL SUMMARY

AI Engineer with robust experience in AI applications development, ranging from architecting Retrieval-Augmented Generation (RAG) systems to deploying Computer Vision models in production. Proficient in PyTorch, with a strong focus on building scalable backend services and LLM applications using FastAPI and Docker. Skilled in optimization techniques for LLMs and developing end-to-end machine learning pipelines that solve complex classification and retrieval problems.

TECHNICAL SKILLS

Machine Learning: Pandas, Numpy, Scikit-Learn, Scipy, seaborn, matplotlib
Deep Learning: PyTorch, TensorFlow/Keras, CNNs, Hugging Face Transformers, LLMs (Gemini, Llama 3).
Generative AI: RAG Architectures, LangChain, Vector Databases (Qdrant), Embeddings.
Backend & MLOps: Python, FastAPI, Docker, REST APIs, Redis, ONNX Runtime.
Computer Vision: Grad-CAM, Image Classification, OpenCV.

PROJECTS

JauapAI – Enterprise RAG System | Python, FastAPI, LangChain, Qdrant, React

- Users required precise, context-aware answers extracted from massive, unstructured academic datasets where traditional search failed.
- Architect and deploy a scalable RAG engine to index and query textbook content in real-time.
- Engineered a high-performance backend using FastAPI. Implemented a Hybrid Search strategy in Qdrant, combining dense embeddings (VoyageAI) with sparse vectors (BGE-M3) for maximum retrieval accuracy. Integrated LlamaParse for complex PDF ingestion and Gemini 3 flash for reasoning.
- Deployed a production-ready MVP that reduced information retrieval time by 90%, delivering grounded answers with high semantic accuracy.

Lung Disease Classification System | Python, TensorFlow/Keras, FastAPI, JavaScript

- Automated analysis of X-ray imagery required a robust, deployable solution to assist in rapid pathology detection.
- Build an end-to-end Computer Vision web application capable of classifying chest X-rays with high sensitivity.
- Trained a custom Convolutional Neural Network (CNN) on the Tuberculosis Chest X-ray dataset using TensorFlow/Keras. Developed a FastAPI microservice to serve model predictions and built a responsive frontend interface. Implemented Grad-CAM (Gradient-weighted Class Activation Mapping) to visualize model attention and improve explainability.
- Successfully deployed a model achieving high recall and precision, providing users with instant visual feedback and confidence scores.

MakeMore – LLM Architecture Implementation | Python, PyTorch, Deep Learning

- Mastering the optimization of Large Language Models requires a first-principles understanding of matrix operations and gradients.
- Build and train autoregressive language models from scratch without high-level abstractions.
- Implemented complex neural architectures including Multi-Layer Perceptrons (MLP), WaveNet, and Decoder-only Transformers. Manually coded Backpropagation, Batch Normalization, and Self-Attention layers to benchmark performance against standard implementations.
- Created convergent models capable of generating coherent text sequences, demonstrating deep technical mastery of Transformer internals.

NLP for Transformers – Model Optimization Pipeline | Python, Hugging Face, ONNX

- Deploying NLP tasks like Named Entity Recognition (NER) in resource-constrained environments required optimized inference.
- Fine-tune and compress pre-trained Transformer models for production use.
- Fine-tuned BERT-based models on custom datasets. Applied Model Quantization and exported models to ONNX format to minimize latency and memory usage.
- Delivered task-specific models that maintained high accuracy while significantly reducing computational overhead.

EDUCATION

Bachelor of Computer Science, KBTU
Self-Study via Books and projects