Key Technical Knowledge For AI Engineers

Basic AI Concepts

- ○ **Machine Learning:** Systems learning patterns from data to improve performance.
  - ■ [High-level intro](), [types of ML](),
- ○ **Deep Learning:** Utilizing neural networks with many layers to model complex patterns.
  - ■ [What is deep learning]()
  - ■ [Math behind deep learning]()
- ○ **Natural Language Processing (NLP):** A class of AIs aimed at processing and understanding human language.
  - ■ **Transformers Architecture**: A specific way to architect AI language models that radically improved quality, by unlocking the AI ability to focus on the right words in the sentences. This concept was first introduced by Google's article 'Attention is All you need'.
    - ● [Transformers, explained]()
    - ● Complicated math behind it if you want to get technical. But it might hurt your brain after watching it. A lot of math…[Transformer Neural Networks, ChatGPT's foundation, Clearly Explained!!!]()
- ○ **Computer Vision:** Interpreting and analyzing visual information from the world.
  - ■ [How computer vision works]()

- ○ **Large Language Models (LLMs):** Large, pre-trained models used as a basis for various AI applications.
  - ■ [Generative AI in Nutshell - how to survive & thrive in the age of AI]()
  - ■ **Tokens:** The numerical representation of text that LLMs read. A token generally corresponds to 4 characters of text for common English text. So based on the size, 100 tokens ~=75 words, using ¾ rule.
    - ● [Understanding ChatGPT/OpenAI Tokens]()
  - ■ **Context Window:** The number of tokens an LLM can receive at every
    - ● [How context window work]()
  - ■ **Hallucinations:** Wrong, made up, answers generated by LLMs
    - ● [Why Large Language Models Hallucinate]()
  - ■ **Prompt Engineering:** Ability to improve the way you ask questions to LLMs so that you get better results.

- - ● [ChatGPT Prompt Engineering mini-course (1h)](#)
    - ● [Mini-course Prompt engineering for vision models](#)
  - ○ [Advanced Prompting Techniques](#)
  - ○ **Top-Players, Tools and Frameworks:** OpenAI, Azure OpenAI, Anthropic, Hugging Face, Langchain, CrewAI, OpenAI's GPTs
  - ○ **Langchain Agents:** AI bots able to use tools that allow them to access APIs, databases and more.
    - ■ [Langchain agents simply explained](#)
  - ○ **Retrieval-Augmented Generation (RAG)**:
    - ■ No Code: Build a RAG System Using Claude 3 Opus And MongoDB
  - ○ **Conversational Agents:** Group of AI agents able to assume roles and plan/discuss/validate answer before providing them.
    - ■ [Here is a lecture from Andrew Ng demonstrating the Agentic Workflows and why they are game-changer](#)
- ● [*FastAPI*](#) = *Web framework for building APIs with Python based on standard Python type hints.*
- ● [*Docker for DataScientists*](#)

Advanced AI Concepts

[**Machine Learning Master**](#) = Great source for complex concepts
- ○ Overfitting and underfitting
  - ■ [Underfitting & Overfitting - Explained](#)
  - ■ [Machine Learning Fundamentals: Bias and Variance](#)
- ○ MLOps [What is MLOps?](#)
- ○ [MLOPS ZoomCamp](#) : 9-Week Course on Productionizing ML Services
- ○ Feature Engineering [What is Feature Engineering?](#)
- ○ Hyperparameter Tunning [Parameters vs hyperparameters in machine learning](#)
- ○ Model Deployment [Machine Learning Model Deployment Explained | All About ML Model Deployment](#)
- ○ Framing the AI task
  - ■ Data cleaning in production time https://youtu.be/P8ERBy91Y90?si=yZ37-ScTbjtK4Af
  - ■ Sampling Frequency
  - ■ Supervised: Classification, Prediction, Regression, Recommendation, Named Entity Recognition (NER), Speech Recognition, Object Detection, Segmentation
    - ● [Image classification vs Object detection vs Image Segmentation | Deep Learning Tutorial 28](#)
    - ● [Natural Language Processing In 5 Minutes | What Is NLP And How Does It Work? | Simplilearn](#)

- - - Unsupervised: Clustering, Dimensionality Reduction
      - [Dimensionality Reduction](#)
      - [Machine Learning - Dimensionality Reduction - Feature Extraction & Selection](#)
    - Reinforcement Learning
      - [OpenAI Plays Hide and Seek…and Breaks The Game! 🤖](#)
    - Generative Models
      - [What are GANs (Generative Adversarial Networks)?](#)
    - Anomaly Detection
  - RLHF (Reinforced Learning with Human Feedback)
    - [Basic explanation](#)
    - If you need to go deeper, [this course](#) is the best
  - Model Distillation
    - [Distilling Neural Networks | Two Minute Papers #218](#)
    - [Quantization vs Pruning vs Distillation: Optimizing NNs for Inference](#)
  - Self-hosting LLMs
    - [All You Need To Know About Running LLMs Locally](#)
  - Fine-tuning
    - [Fine-tuning a Neural Network explained](#)
    - [Catastrophic forgetting](#)

Agents- Advanced AI Concepts

  - [4 AI Agent Strategies](#): (DeepLearning.AI)
    - [Reflection](#)
      - LangChain - [Reflection Agents example](#)
    - [Tool Use](#)
      - LangChain - [Tool Calling](#)
    - [Planning](#)
      - LangGraph - [Planning Agents](#)
    - [Multi-agent Collaboration](#)
      - LangGraph: [Multi-Agent Workflows](#)

    - Agent Evaluations
      - [LangGraph: Agent Evaluations](#)
      - [LangSmith Without LangChain/Graph (Observability, Evals)](#)

    - Designing with Agents in Mind *(Important for Frontend developers)*
      - [The Agentic Era of UX](#)
      - [LangChain: Breakout Agentic Apps](#)
      - [LangChain: UX for Agents, Part 1: Chat](#)
      - [LangChain: UX for Agents, Part 2: Ambient](#)
      - [LangChain: UX for Agents, Part 3: Spreadsheets, Generative, and Collaborative UI/UX](#)

Examples of Developing AI Products

[Create ChatGPT Application with Chat GPT API from OpenAI in Python Flask](#)

[Build AI Apps with ChatGPT and DALL-E](#)

[Developer channel](#)

Introduction to Databases

Database Basic:
[Microsoft database foundations.](#)

SQL Basics and Advanced Queries
- (58mins) [Learn SQL in 1 hour](#)
- (docs) [SQL reference](#)

Database Design and Normalization
Basic lesson:
- (17mins) [Database design tutorial.](#)
- (6mins) [Database schema](#)
- (32mins) [Logical Database Design and E-R Diagrams](#)
- (docs) [The design process](#)
- (docs) [Applying the normalization rules](#)

Advanced lesson: (optional)
- (27 hours) [Meta Intro to database.](#)

Python Fundamentals
- (3h55mins) [Python for beginners](#)

Data Manipulation using Python (Pandas, NumPy)
- (1h49min) [Python Library for Data Science](#)

Connecting Python to Databases (SQLAlchemy, psycopg2)

- (8mins) Python database Connection
- (17mins) SQL database with Pandas and Python