

# Ultimate Roadmap to Build an LLM

## Phase 1 – Foundations (Math + Programming)

You must be solid here.

- **Python Programming:** Data structures, OOP, NumPy, pandas, matplotlib
- **Math for ML:**
  - Linear Algebra → *Gilbert Strang MIT Course* (YouTube)
  - Calculus for ML → *Khan Academy or 3Blue1Brown videos*
  - Probability & Statistics → *StatQuest* (YouTube)

### Resources:

- Python: [freeCodeCamp Python Course](#)
  - Linear Algebra: MIT 18.06 – Gilbert Strang
  - Probability: [StatQuest Playlist](#)
- 

## Phase 2 – Machine Learning Basics

Learn the ML pipeline, supervised vs unsupervised, train/test splits, metrics, etc.

### Resources:

- Andrew Ng's ML Course (Coursera)
- Book: *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow* by Aurélien Géron

Practice:

- Kaggle beginner competitions
  - Build simple projects (spam detection, sentiment analysis)
- 

## Phase 3 – Deep Learning Fundamentals

Understand neural networks deeply (how forward/backprop works, gradients, optimizers).

### Resources:

- DeepLearning.ai – Deep Learning Specialization (Coursera)
- *Dive into Deep Learning* (free book): d2l.ai
- *fast.ai Practical Deep Learning for Coders*: fast.ai course

Practice:

- Implement a neural network from scratch (NumPy)
  - Train CNN on MNIST/CIFAR
-

## Phase 4 – NLP + Transformers

Here you step into modern LLM building blocks.

- Learn embeddings, attention, seq2seq, BERT, GPT
- Understand the **Transformer architecture** fully

### Resources:

- Blog: *The Illustrated Transformer* by Jay Alammar
- Hugging Face Course (free): [huggingface.co/course](https://huggingface.co/course)
- Paper: *Attention Is All You Need* (original transformer paper)

Practice:

- Use Hugging Face Transformers to fine-tune BERT/GPT-2 on your own dataset
  - Build small chatbot using pretrained models
- 

## Phase 5 – Build GPT-like Models from Scratch

Start small:

- Learn tokenization
- Train a tiny GPT (like nanoGPT)
- Understand training infrastructure (GPUs/TPUs)

### Resources:

- Andrej Karpathy's **nanoGPT**: [github.com/karpathy/nanoGPT](https://github.com/karpathy/nanoGPT)
- His video *"Let's build GPT from scratch"* (YouTube)

Practice:

- Implement a small transformer from scratch in PyTorch
  - Train on a small dataset (Shakespeare text, Wikipedia snippets)
- 

## Phase 6 – Scaling Up (LLM Engineering)

To make something like ChatGPT:

- Learn distributed training (Data Parallel, Model Parallel)
- Learn about **LoRA**, **quantization**, **fine-tuning**
- Use open models like LLaMA, Falcon, Mistral as a base
- Use **LangChain** or **Haystack** for building apps on top of LLMs

### Resources:

- *LLM Bootcamp* by Full Stack Deep Learning
- *Efficient Fine-Tuning of LLMs* (LoRA paper + tutorials)
- LangChain Docs: [docs.langchain.com](https://docs.langchain.com)

---

## Phase 7 – Deployment + Serving

Once your model is trained:

- Learn how to deploy models using APIs (FastAPI/Flask)
- Optimize inference with ONNX, quantization
- Build a frontend (React/Next.js) to interact with your model

### Resources:

- FastAPI Docs
- Hugging Face Inference API
- *Deploying ML Models* by Full Stack Deep Learning

---

## Phase 8 – Practice / Portfolio Projects

- Fine-tune GPT-2 for custom chatbot
- Text summarizer for news articles
- AI tutor app using LangChain
- Build a retrieval-augmented generation (RAG) pipeline with your own data

---

### Summary Roadmap Table

Phase Goal		Resource
1	Math & Python	MIT, StatQuest
2	ML Basics	Andrew Ng ML
3	Deep Learning	DeepLearning.ai, D2L
4	NLP + Transformers	Hugging Face Course
5	Build Tiny GPT	nanoGPT
6	Scale Up + Fine-Tune	LLaMA, LoRA tutorials
7	Deploy LLM	FastAPI, LangChain
8	Portfolio Projects	Kaggle, GitHub

---

### Tips for Success

- **Don't jump directly to LLMs.** Build the foundations first.
- **Implement things from scratch** (even tiny versions) to really learn.
- **Use GPUs** (Google Colab, Kaggle Notebooks, or rent on Paperspace/RunPod).
- **Join communities** (Hugging Face forums, Reddit ML, Discord AI groups).