



3-6 Month ML Engineer Career Roadmap

From Junior to Job-Ready



PHASE 1: FOUNDATION STRENGTHENING (Weeks 1-4)

Week 1-2: Deep Learning Fundamentals

Daily Schedule (3-4 hours/day):

Morning (1.5 hours):

- Fast.ai Practical Deep Learning Course (Lesson 1-4)
- OR Andrew Ng's Deep Learning Specialization (Course 1: Neural Networks)
- Focus: Understanding neural networks, backpropagation, gradient descent

Evening (1.5 hours):

- Implement from scratch:
 - Simple neural network for MNIST
 - Activation functions (ReLU, Sigmoid, Softmax)
 - Basic gradient descent optimizer
- Use only NumPy first, then compare with PyTorch

Weekend Project: Build a digit recognizer (MNIST) with PyTorch:

- Create clean code structure
- Add data augmentation
- Achieve 98%+ accuracy
- Deploy on Hugging Face Spaces
- Write detailed README

Resources:

- Fast.ai: <https://course.fast.ai/>
- PyTorch tutorials: <https://pytorch.org/tutorials/>

- Papers with Code: <https://paperswithcode.com/>
-

Week 3-4: Deep Learning Framework Mastery (PyTorch)

Daily Focus:

Technical Skills (2 hours):

- PyTorch Lightning (cleaner code structure)
- Custom datasets and data loaders
- Transfer learning concepts
- Model checkpointing and saving
- Tensorboard for visualization

Practice (1.5 hours):

- Kaggle competitions (participate, don't aim to win):
 - Join one active competition
 - Study winning solutions from past competitions
 - Understand ensemble techniques

Weekend Project: Image Classification with Transfer Learning:

- Use ResNet/EfficientNet on custom dataset
- Fine-tune pre-trained models
- Create inference API with FastAPI
- Dockerize the application
- Deploy on Railway/Render

Deliverables:

- 2 complete GitHub projects with deployment
 - Blog post: "Understanding Transfer Learning"
-

PHASE 2: SPECIALIZATION & REAL PROJECTS (Weeks 5-10)

Week 5-7: Choose ONE Specialization & Deep Dive

Option A: Computer Vision

Week 5:

- Object Detection (YOLO, Faster R-CNN)
- Image Segmentation basics
- Build: Real-time object detector using webcam

Week 6:

- Advanced architectures (Vision Transformers)
- Data augmentation strategies (Albumentations)
- Build: Face detection + emotion recognition system

Week 7:

- OCR and document analysis
- Build: Document scanner + text extractor app
- Deploy with Streamlit

Option B: Natural Language Processing

Week 5:

- Transformers architecture deep dive
- Hugging Face ecosystem mastery
- Build: Sentiment analysis API for product reviews

Week 6:

- Fine-tuning BERT/RoBERTa
- Text classification and NER
- Build: News article classifier + entity extractor

Week 7:

- Text generation (GPT-style models)

- Build: Custom chatbot using RAG (Retrieval Augmented Generation)
- Deploy with LangChain + FastAPI

Daily Routine (4 hours):

- 2 hours: Learn new concepts + tutorials
 - 1.5 hours: Implement learned concepts
 - 30 mins: Document and write tests
-

Week 8-10: Industry-Grade Project (Portfolio Centerpiece)

Goal: Build ONE production-ready ML application that shows end-to-end skills

Project Ideas by Specialization:

Computer Vision:

- Smart Parking System (vehicle detection + space monitoring)
- Medical Image Analysis (X-ray/CT scan classifier)
- Product Quality Inspector (defect detection)

NLP:

- Customer Support Chatbot (with RAG)
- Resume Parser + Job Matcher
- Fake News Detector with explanation

Project Requirements (NON-NEGOTIABLE):

1. Real problem solving (not just tutorial copy)
2. Clean, modular code with proper structure
3. Unit tests for critical functions
4. CI/CD pipeline (GitHub Actions)
5. Dockerized application
6. Deployed and accessible via URL
7. Monitoring/logging implemented

8. Professional README with architecture diagram
9. API documentation (Swagger/OpenAPI)
10. Demo video (2-3 minutes)

Weekly Breakdown:

Week 8: Planning & Data

- Define problem and success metrics
- Collect/create dataset (min 1000+ samples)
- Exploratory Data Analysis
- Set up project structure and GitHub repo

Week 9: Model Development

- Baseline model implementation
- Experiment tracking (MLflow/Weights & Biases)
- Model optimization and tuning
- Evaluate multiple architectures

Week 10: Deployment & Polish

- Create FastAPI/Flask backend
- Add caching (Redis) for better performance
- Containerize with Docker
- Deploy on cloud (AWS/GCP/Railway)
- Add monitoring and error handling
- Write comprehensive documentation

PHASE 3: MLOps & SYSTEM DESIGN (Weeks 11-14)

Week 11-12: MLOps Fundamentals

Core Skills to Learn:

Day 1-3: Experiment Tracking

- MLflow basics: tracking, projects, models
- Weights & Biases integration
- Build: Retrain your previous project with full experiment tracking

Day 4-6: Model Versioning & Registry

- DVC (Data Version Control)
- Model registries and versioning
- Build: Version control your datasets and models

Day 7-10: CI/CD for ML

- GitHub Actions for ML workflows
- Automated testing for ML code
- Automated model retraining pipeline
- Build: CI/CD pipeline for one of your projects

Day 11-14: Monitoring & Observability

- Model performance monitoring
- Data drift detection
- Prometheus + Grafana basics
- Build: Add monitoring dashboard to your main project

Weekend Deep Dive: Read and implement: "Machine Learning System Design" patterns

- Study: Netflix, Spotify ML systems (blog posts)
- Understand: Model serving architectures
- Practice: Design a recommendation system on paper

Week 13-14: Cloud Platforms & Scalability

Week 13: AWS/GCP Basics

Must Learn Services:

- AWS: EC2, S3, Lambda, SageMaker (free tier)

- OR GCP: Compute Engine, Cloud Storage, Vertex AI
- Kubernetes basics (Minikube locally)

Hands-on Tasks:

- Deploy model on AWS Lambda (serverless)
- Set up S3 for data storage
- Create EC2 instance and deploy your app
- Use RDS for database needs

Week 14: Scalability & Performance

- Load balancing concepts
- Caching strategies (Redis)
- Batch vs real-time inference
- Optimize model for production (quantization, pruning)

Major Exercise: Take your main project and:

- Make it handle 1000+ requests/second (load testing)
- Optimize inference time (< 100ms if possible)
- Add auto-scaling capabilities
- Create cost analysis document

PHASE 4: JOB PREPARATION (Weeks 15-20)

Week 15-16: Resume, Portfolio & Personal Brand

Week 15: Professional Portfolio

Tasks:

1. Create professional portfolio website:
 - Use: React + Tailwind OR Hugo/Jekyll
 - Sections: About, Projects, Skills, Blog, Contact
 - Deploy: Vercel/Netlify (free)

2. Polish GitHub Profile:

- Professional README with stats
- Pin best 6 projects
- Ensure all projects have excellent READMEs
- Add topics/tags to repositories

3. Write 3-4 Technical Blog Posts:

- "Building [Your Project]: A Complete Guide"
- "5 Mistakes I Made in My First ML Project"
- "How I Deployed ML Model to Production"
- "Understanding [Specific ML Concept] with Code"

Week 16: Resume & LinkedIn

Resume (ATS-friendly):

- Format: Single column, clean, PDF
- Sections: Summary, Skills, Projects, Experience, Education
- Quantify everything (improved accuracy by X%, reduced latency by Y ms)
- Use action verbs: Built, Deployed, Optimized, Implemented
- Tailor for ML Engineer role

LinkedIn Optimization:

- Professional photo
 - Compelling headline: "ML Engineer | Python | PyTorch | Building AI Solutions"
 - Detailed work/project descriptions
 - Add all your projects
 - Post weekly about your learning journey
 - Connect with ML engineers and recruiters
-

Week 17-18: Interview Preparation - Technical

Daily Schedule (3-4 hours):

Morning (1 hour): LeetCode/HackerRank

- Focus: Medium difficulty problems
- Topics: Arrays, Strings, Trees, Dynamic Programming
- Target: 2-3 problems daily
- Keep a log of patterns learned

Afternoon (1.5 hours): ML Theory Study these topics thoroughly:

1. ML Fundamentals:

- Bias-variance tradeoff
- Overfitting/underfitting
- Cross-validation techniques
- Evaluation metrics (precision, recall, F1, ROC-AUC)

2. Deep Learning:

- How backpropagation works
- Different optimizers (SGD, Adam, RMSprop)
- Batch normalization, dropout
- Common architectures and why they work

3. MLOps:

- Model deployment strategies
- A/B testing
- Model monitoring
- Handling data drift

Evening (1 hour): ML System Design Practice designing systems on paper:

- "Design a recommendation system for e-commerce"
- "Design a fraud detection system"
- "Design an image search engine"
- "Design a real-time translation service"

Framework for answers:

1. Clarify requirements
2. Define success metrics
3. Design data pipeline
4. Choose model architecture
5. Discuss training strategy
6. Explain deployment approach
7. Address monitoring and maintenance

Weekend: Mock Interviews

- Use Pramp.com (free peer interviews)
 - Practice with friends
 - Record yourself and review
-

Week 19-20: Interview Preparation - Behavioral & Applications

Week 19: Behavioral Interview Prep

Prepare STAR Stories for:

- Tell me about yourself (2-minute pitch)
- Challenging project you worked on
- Time you failed and what you learned
- Conflict resolution
- Why ML engineering?
- Why this company?
- Where do you see yourself in 5 years?

Practice Questions:

- "Explain [your project] to a non-technical person"
- "What's the most interesting ML paper you've read recently?"
- "How do you stay updated with ML trends?"

Daily Practice:

- Record yourself answering questions
- Practice with different styles (enthusiastic, concise, detailed)
- Time yourself (answers should be 1-3 minutes)

Week 20: Job Applications Blitz

Application Strategy:

Where to Apply:

1. Job Boards:

- LinkedIn (filter: Entry-level, ML Engineer)
- Indeed, Glassdoor
- AngelList (startups)
- RemoteML.com
- AI-Jobs.net

2. Company Types:

- Startups (more flexible with junior candidates)
- Tech consultancies
- Product companies with ML teams
- Research labs hiring engineers

3. Bangladesh-Specific:

- bdjobs.com, Prothom-Alo Jobs
- Local startups: Pathao, Shohoz, Chaldal (tech teams)
- International remote positions

Daily Target:

- 5-10 job applications
- 2-3 cold emails to hiring managers
- Connect with 5-10 ML engineers on LinkedIn

Application Customization:

- Read job description carefully

- Match your resume to key requirements
 - Write personalized cover letter highlighting relevant projects
 - Follow up after 1 week
-

DAILY ROUTINE TEMPLATE

Weekdays (Assuming 4-5 hours study time):

Morning (2 hours) - 6:00 AM - 8:00 AM:

- 1 hour: Course/Tutorial (new concepts)
- 1 hour: LeetCode/DSA practice

Evening (2.5 hours) - 7:00 PM - 9:30 PM:

- 1.5 hours: Hands-on coding (projects)
- 30 mins: Read ML papers/blogs
- 30 mins: Documentation/GitHub updates

Before Bed (30 mins):

- Review Anki flashcards (ML concepts)
- LinkedIn activity (post/comment)

Weekends (6-8 hours):

Saturday: Deep Work on Projects

- 4-6 hours: Major project development
- 2 hours: Learning new tools/frameworks

Sunday: Review & Plan

- Morning: Blog writing
 - Afternoon: Code review of your own projects
 - Evening: Plan next week's learning goals
-

Courses (Pick based on your pace):

1. **Fast.ai Practical Deep Learning** (Free, highly practical)
2. **Full Stack Deep Learning** (Free, focuses on production)
3. **Made with ML** (Free, MLOps focused)
4. **DeepLearning.AI TensorFlow/PyTorch courses**

Books (One at a time):

1. **"Designing Machine Learning Systems"** by Chip Huyen (MUST READ)
2. **"Machine Learning Engineering"** by Andriy Burkov
3. **"Deep Learning"** by Goodfellow (reference book)

YouTube Channels:

- Andrej Karpathy (Neural Networks from scratch)
- StatQuest (ML concepts explained simply)
- Two Minute Papers (stay updated)
- Yannic Kilcher (paper explanations)

Communities to Join:

- r/MachineLearning (Reddit)
- MLOps Community (Slack)
- Kaggle forums
- Discord servers for PyTorch/TensorFlow

Tools to Master:

- **Version Control:** Git, GitHub
- **ML Frameworks:** PyTorch (primary), TensorFlow (basic)
- **MLOps:** MLflow, Weights & Biases
- **Deployment:** Docker, FastAPI, Streamlit
- **Cloud:** AWS/GCP basics
- **Monitoring:** Prometheus, Grafana
- **Databases:** PostgreSQL, MongoDB basics

KEY MILESTONES & CHECKPOINTS

End of Month 1:

- ✓ 2 deployed ML projects
- ✓ Solid PyTorch fundamentals
- ✓ Active GitHub profile
- ✓ Started blogging

End of Month 2:

- ✓ 1 industry-grade project (portfolio centerpiece)
- ✓ Specialization chosen and progressing
- ✓ Basic MLOps knowledge
- ✓ 3-4 technical blog posts

End of Month 3:

- ✓ Total 4-5 impressive projects
- ✓ MLOps skills demonstrated
- ✓ Professional portfolio live
- ✓ Strong understanding of ML system design

End of Month 4-5:

- ✓ Interview-ready (technical + behavioral)
- ✓ 50+ job applications sent
- ✓ Active networking and LinkedIn presence
- ✓ Confident explaining all your projects

Month 6:

- ✓ Continuing applications and interviews
- ✓ Negotiating offers
- ✓ Landed first ML role! 🎉

COMMON PITFALLS TO AVOID

1. Tutorial Hell

- ✗ Watching course after course without building
- ✓ Learn concept → Implement immediately → Build something unique

2. Perfection Paralysis

- ✗ Waiting until project is "perfect" before sharing
- ✓ Ship early, iterate based on feedback

3. Breadth over Depth

- ✗ Learning 10 things superficially
- ✓ Master 3-4 things deeply

4. Ignoring Fundamentals

- ✗ Jumping to advanced topics without solid basics
- ✓ Ensure strong foundation before advancing

5. No Documentation

- ✗ Code without README or comments