

Complete AI Mastery Course: From Foundations to Advanced Applications

A comprehensive course designed for learners with ML/DL experience who want to master modern AI

Course Overview & Learning Path

Duration: 12-16 weeks (flexible, self-paced)

Prerequisites: Machine Learning and Deep Learning fundamentals

Goal: Master modern AI systems, from transformers to AI agents

Module 1: Modern AI Foundations (Week 1-2)

1.1 AI vs ML vs DL: The Modern Landscape

- **Conceptual Framework**
 - Artificial Intelligence: The broad goal of machine intelligence
 - Machine Learning: Statistical methods for learning from data
 - Deep Learning: Neural networks with multiple layers
 - Modern AI: Transformer-based, multimodal, agent-capable systems
- **Key Paradigm Shifts**
 - From rule-based to statistical learning
 - From feature engineering to representation learning
 - From task-specific to foundation models
 - From prediction to generation and reasoning

1.2 The Transformer Revolution

- **Architecture Deep Dive**
 - Self-attention mechanism mathematics
 - Multi-head attention implementation
 - Positional encoding strategies
 - Layer normalization vs batch normalization
- **Hands-on Project:** Build a mini-transformer from scratch

python

```
# Implement attention mechanism
def scaled_dot_product_attention(Q, K, V, mask=None):
    # Your implementation here
    pass

# Build transformer block
class TransformerBlock(nn.Module):
    # Your implementation here
    pass
```

1.3 Foundation Models & Transfer Learning

- **Pre-training Paradigms**
 - Autoregressive language modeling (GPT-style)
 - Masked language modeling (BERT-style)
 - Encoder-decoder models (T5-style)
 - Multimodal pre-training
- **Fine-tuning Strategies**
 - Full fine-tuning vs parameter-efficient methods
 - LoRA (Low-Rank Adaptation)
 - Prefix tuning and prompt tuning
 - In-context learning and few-shot prompting

Practice Exercise: Fine-tune a pre-trained model for your domain

Module 2: Large Language Models (Week 3-4)

2.1 LLM Architecture & Training

- **Model Architectures**
 - GPT family (GPT-1 through GPT-4)
 - PaLM, LLaMA, Mistral architectures
 - Mixture of Experts (MoE) models
 - State Space Models (Mamba)
- **Training Pipeline**
 - Data collection and preprocessing

- Tokenization strategies (BPE, SentencePiece)
- Distributed training techniques
- Gradient accumulation and mixed precision

2.2 Prompt Engineering Mastery

- **Fundamental Techniques**
 - Zero-shot, one-shot, few-shot learning
 - Chain of thought prompting
 - Tree of thoughts reasoning
 - Self-consistency decoding
- **Advanced Prompting**
 - Role-based prompting
 - Constitutional AI principles
 - Prompt chaining and decomposition
 - Adversarial prompt testing

Hands-on Lab: Build a prompt optimization framework

```
python

def optimize_prompt(base_prompt, test_cases, model):
    """
    Systematically improve prompts using A/B testing
    """
    # Your implementation
    pass
```

2.3 LLM Evaluation & Alignment

- **Evaluation Metrics**
 - Perplexity and likelihood measures
 - BLEU, ROUGE, BERTScore
 - Human evaluation protocols
 - Benchmark suites (HELM, BIG-bench)
- **Alignment Techniques**
 - Supervised fine-tuning (SFT)

- Reinforcement Learning from Human Feedback (RLHF)
- Constitutional AI
- Direct Preference Optimization (DPO)

Project: Implement a simple RLHF pipeline

Module 3: Multimodal AI Systems (Week 5-6)

3.1 Vision-Language Models

- **Architecture Patterns**
 - CLIP: Contrastive Language-Image Pre-training
 - DALL-E and diffusion models
 - GPT-4V and multimodal transformers
 - Flamingo and few-shot multimodal learning
- **Technical Deep Dive**
 - Cross-modal attention mechanisms
 - Vision transformers (ViTs)
 - Contrastive learning objectives
 - Multimodal fusion strategies

3.2 Generative AI: Text, Image, and Beyond

- **Diffusion Models**
 - DDPM mathematics and implementation
 - Stable Diffusion architecture
 - ControlNet and conditioning
 - Video generation (RunwayML, Pika)
- **Audio Generation**
 - Speech synthesis (WaveNet, Tacotron)
 - Music generation (MuseNet, Jukebox)
 - Voice cloning techniques
 - Real-time audio processing

Creative Project: Build a multimodal content generator

python

```
class MultimodalGenerator:
    def __init__(self):
        self.text_model = GPT()
        self.image_model = StableDiffusion()
        self.audio_model = WaveNet()

    def generate_story_with_media(self, prompt):
        # Generate text, images, and audio together
        pass
```

3.3 Multimodal Applications

- **Document AI**
 - OCR and document understanding
 - Layout analysis and information extraction
 - Table and chart parsing
 - Legal and medical document processing
 - **Embodied AI**
 - Robotics and computer vision
 - Autonomous vehicle perception
 - Augmented/Virtual reality
 - Human-computer interaction
-

Module 4: AI Agents & Reasoning Systems (Week 7-8)

4.1 AI Agent Architectures

- **Agent Frameworks**
 - ReAct (Reasoning and Acting)
 - Plan-and-Execute agents
 - Multi-agent systems
 - Tool-using AI (Toolformer)
- **Memory Systems**
 - Short-term vs long-term memory

- Vector databases and retrieval
- Episodic memory architectures
- Memory consolidation strategies

4.2 Advanced Reasoning

- **Symbolic Reasoning Integration**
 - Neuro-symbolic AI approaches
 - Knowledge graphs and reasoning
 - Logical inference engines
 - Constraint satisfaction
- **Planning and Problem Solving**
 - Classical AI planning algorithms
 - Hierarchical task networks
 - Monte Carlo Tree Search integration
 - Game-theoretic reasoning

Capstone Project: Build an AI agent for complex task completion

```
python

class AdvancedAIAgent:
    def __init__(self):
        self.llm = LanguageModel()
        self.tools = ToolRegistry()
        self.memory = VectorMemory()
        self.planner = TaskPlanner()

    def execute_complex_task(self, goal):
        plan = self.planner.create_plan(goal)
        for step in plan:
            result = self.execute_step(step)
            self.memory.store(step, result)
        return self.synthesize_results()
```

4.3 Retrieval-Augmented Generation (RAG)

- **RAG Architectures**
 - Dense retrieval with DPR

- Sparse retrieval with BM25
 - Hybrid retrieval systems
 - Iterative retrieval and generation
 - **Advanced RAG Techniques**
 - Query expansion and rewriting
 - Multi-hop reasoning
 - Citation and attribution
 - Knowledge base construction
-

Module 5: AI Infrastructure & MLOps (Week 9-10)

5.1 Model Deployment & Serving

- **Inference Optimization**
 - Model quantization techniques
 - Pruning and distillation
 - ONNX and TensorRT optimization
 - Hardware acceleration (TPUs, custom chips)
- **Serving Infrastructure**
 - Model serving frameworks (Triton, TorchServe)
 - Load balancing and auto-scaling
 - A/B testing for models
 - Monitoring and observability

5.2 Training Infrastructure

- **Distributed Training**
 - Data parallelism vs model parallelism
 - Pipeline parallelism
 - Gradient compression techniques
 - Fault tolerance and checkpointing
- **Cloud Platforms**
 - AWS SageMaker deep dive
 - Google Vertex AI workflows

- Azure ML pipelines
- Multi-cloud strategies

Infrastructure Project: Deploy a production AI system

```
yaml

# kubernetes deployment example
apiVersion: apps/v1
kind: Deployment
metadata:
  name: ai-model-serving
spec:
  replicas: 3
  selector:
    matchLabels:
      app: ai-model
  template:
    spec:
      containers:
        - name: model-server
          image: your-ai-model:latest
          resources:
            requests:
              nvidia.com/gpu: 1
```

5.3 Data Engineering for AI

- **Data Pipeline Design**
 - ETL for unstructured data
 - Real-time data processing
 - Data quality and validation
 - Privacy-preserving techniques
 - **Vector Databases**
 - Pinecone, Weaviate, Chroma
 - Embedding generation and storage
 - Similarity search optimization
 - Index management strategies
-

Module 6: Specialized AI Applications (Week 11-12)

6.1 AI for Science & Research

- **Scientific Computing**
 - Protein folding (AlphaFold)
 - Drug discovery pipelines
 - Climate modeling acceleration
 - Automated theorem proving
- **Research Acceleration**
 - Literature review automation
 - Hypothesis generation
 - Experimental design optimization
 - Scientific writing assistance

6.2 Enterprise AI Solutions

- **Industry Applications**
 - Financial services (fraud detection, algorithmic trading)
 - Healthcare (diagnosis, drug discovery)
 - Manufacturing (predictive maintenance, quality control)
 - Retail (recommendation systems, demand forecasting)
- **Business Integration**
 - ROI measurement for AI projects
 - Change management for AI adoption
 - Compliance and regulatory considerations
 - Stakeholder communication strategies

Industry Project: Choose your domain and build an end-to-end solution

6.3 Creative AI & Human-AI Collaboration

- **Creative Applications**
 - Content generation workflows
 - Art and design automation
 - Music composition systems

- Game development AI
 - **Human-AI Interaction**
 - Interface design principles
 - Explainable AI techniques
 - Trust and transparency
 - Augmentation vs automation strategies
-

Module 7: AI Safety, Ethics & Future Trends (Week 13-14)

7.1 AI Safety & Alignment

- **Technical Safety**
 - Robustness and adversarial examples
 - Interpretability techniques
 - Uncertainty quantification
 - Failure mode analysis
- **Alignment Research**
 - Value learning and specification
 - Corrigibility and shutdown problems
 - Mesa-optimization risks
 - Scalable oversight techniques

7.2 Bias, Fairness & Ethics

- **Bias Detection & Mitigation**
 - Statistical parity measures
 - Individual vs group fairness
 - Intersectionality considerations
 - Bias audit frameworks
- **Ethical Frameworks**
 - Consequentialist vs deontological ethics
 - Stakeholder analysis
 - Cultural sensitivity in AI
 - Global AI governance

7.3 Future of AI

- **Emerging Trends**

- Artificial General Intelligence (AGI) pathways
- Quantum machine learning
- Neuromorphic computing
- Brain-computer interfaces

- **Societal Impact**

- Economic transformation scenarios
 - Education and skill evolution
 - Governance and regulation
 - Long-term existential considerations
-

Module 8: Capstone Project & Portfolio (Week 15-16)

8.1 Choose Your Capstone

Select one of these advanced projects:

Option A: Advanced AI Agent

- Build an AI agent that can handle complex, multi-step tasks
- Include planning, tool use, memory, and learning capabilities
- Deploy with proper monitoring and evaluation

Option B: Multimodal AI Application

- Create an application combining text, image, and audio AI
- Focus on user experience and practical utility
- Include real-time processing capabilities

Option C: AI for Domain Expertise

- Pick a specific domain (healthcare, finance, education, etc.)
- Build an AI system that provides expert-level assistance
- Include domain-specific evaluation metrics

Option D: AI Safety/Alignment Research

- Implement and evaluate alignment techniques
- Contribute to open safety research
- Include comprehensive evaluation and analysis

8.2 Portfolio Development

- **Documentation & Presentation**
 - Technical blog posts
 - Code repositories with clear documentation
 - Demo videos and live presentations
 - Peer review and feedback incorporation
 - **Professional Development**
 - GitHub portfolio optimization
 - LinkedIn AI expertise showcase
 - Conference talk preparation
 - Open source contributions
-

Learning Resources & Tools

Essential Tools

- **Development Environment**
 - Python 3.9+ with virtual environments
 - Jupyter notebooks / Google Colab
 - VS Code with AI extensions
 - Git for version control
- **Frameworks & Libraries**

```
bash
```

Core AI libraries

```
pip install torch torchvision transformers  
pip install tensorflow tensorflow-datasets  
pip install huggingface-hub datasets  
pip install langchain openai anthropic
```

Specialized tools

```
pip install faiss-cpu pinecone-client  
pip install streamlit gradio  
pip install wandb tensorboard  
pip install accelerate deepspeed
```

Recommended Reading

• Papers

- "Attention Is All You Need" (Transformer paper)
- "Language Models are Few-Shot Learners" (GPT-3)
- "Training language models to follow instructions" (InstructGPT)
- "Constitutional AI: Harmlessness from AI Feedback"

• Books

- "Deep Learning" by Ian Goodfellow
- "The Alignment Problem" by Brian Christian
- "Human Compatible" by Stuart Russell
- "Superintelligence" by Nick Bostrom

Online Communities

• Research Communities

- Papers With Code
- Hugging Face Hub
- AI Alignment Forum
- ML Twitter community

• Practical Communities

- Stack Overflow AI tags
- Reddit r/MachineLearning
- Discord AI servers