PRIMEX AI MASTERY DOCTRINE v1.0

Commander Objective: Achieve operational mastery across AI theory, engineering, systems, leadership, and governance. Capable of instructing advanced learners and commanding elite AI teams.

---

0) OUTCOMES & STANDARDS

- **Operator Level:** Design, build, deploy, and evaluate AI systems end-to-end; justify choices mathematically and empirically.
- **Instructor Level:** Teach fundamentals → frontier methods; design labs; evaluate students objectively.
- **Commander Level:** Architect multi-model systems; lead teams; enforce MLOps, safety, and governance; set research/roadmap strategy.

Mastery Checks (Representative): - Derive gradients for logistic regression and implement SGD from scratch. - Re-derive Transformer self-attention; implement a minimal GPT-like model; train on a small corpus; run evals. - Implement retrieval-augmented generation (RAG) with vector search; measure grounding, hallucination rate, and latency. - Quantize a model to INT8/INT4; benchmark throughput/latency; document tradeoffs. - Design and run offline + online evals with statistical power analysis; ship A/B safely. - Threat model prompt-injection/data exfiltration; build guardrails and red-team playbooks.

---

1) CORE FOUNDATIONS (NON-NEGOTIABLE)

Mathematics - Linear Algebra: vectors, matrices, eigen/singular values, matrix calculus. - Probability & Statistics: random variables, expectations, MLE/MAP, Bayesian inference, hypothesis testing, confidence/credible intervals. - Optimization: convexity, gradients, SGD variants, Adam, second-order intuition, constraints, Lagrangian.

Computer Science - Data structures & algorithms; complexity; numerical stability; floating point. - Systems: OS/containers/networking; GPU basics (CUDA/compute/memory hierarchy); distributed systems (RPC, queues, consensus basics). - Python proficiency; idiomatic PyTorch/JAX; vectorization; mixed precision.

---

2) MACHINE LEARNING FUNDAMENTALS

- Supervised learning: bias–variance, regularization, cross-validation, calibration.
- Models: linear/logistic regression, naïve Bayes, SVMs, trees/ensembles (RF/GBMs/XGBoost/LightGBM), kNN, k-means, PCA/TSNE/UMAP.
- Feature engineering & leakage prevention; pipelines; data versioning.
- Evaluation: metrics per task (AUC/PR, F1, logloss, RMSE), cost-sensitive metrics, stratification, imbalanced data.

Mastery Drills: - Implement logistic regression + softmax regression from scratch; compare to scikit-learn. - Train/benchmark gradient boosting vs. small MLP on tabular task; explain wins/losses.

---

3) DEEP LEARNING CORE

- Neural Networks: MLPs, CNNs, RNNs (LSTM/GRU), attention mechanisms.
- Training: initialization, normalization (Batch/Layer/RMSNorm), regularization (dropout, weight decay), learning rate schedules, early stopping.
- Transformers: attention math, positional encodings, encoder/decoder/seq2seq, masking, causal LM vs. encoder-only.
- Optimization pathologies: vanishing/exploding gradients; gradient clipping; loss spikes; FP16/BF16.

Mastery Drills: - Build a mini-Transformer (PyTorch) that learns toy language modeling. - Visualize attention maps; show how context length impacts perplexity & memory.

---

4) NATURAL LANGUAGE PROCESSING (NLP)

- Tokenization (BPE/WordPiece/Unigram); vocab design; byte-level tradeoffs.
- Pretraining objectives: MLM, CLM, seq2seq; scaling laws; data curation.
- Fine-tuning strategies: supervised finetuning (SFT), LoRA/QLoRA, instruction tuning, RLHF/RLAIF.
- Retrieval-Augmented Generation (RAG): index selection (FAISS, HNSW), chunking, hybrid search (sparse+dense), query rewriting, reranking.
- Evaluation: perplexity vs. task metrics; human evals; rubrics; hallucination detection; grounding score.

Mastery Drills: - Build RAG over a small doc set; implement hybrid retrieval; run ablations on chunk size, overlap, and reranker. - Fine-tune a small open LLM with LoRA on a domain dataset; measure instruction-following gains.

---

5) REINFORCEMENT LEARNING (RL)

- MDPs, value/policy methods; Q-learning; policy gradients; actor-critic; exploration vs. exploitation.
- Offline RL; contextual bandits; safe RL; reward modeling for LLMs; PPO/DPO/GRPO-style methods.
- Multi-agent coordination; self-play; curriculum learning.

Mastery Drills: - Train PPO on a toy environment; add reward shaping; evaluate stability and sample efficiency.

---

6) VISION, AUDIO, MULTIMODAL

- CNNs → Vision Transformers (ViT); detection/segmentation (Faster/Mask R-CNN, DETR); diffusion models for images/audio/video.

- CLIP-style contrastive learning; vision-language pretraining (BLIP/ALBEF); grounding & perception-action loops.

Mastery Drills: - Fine-tune a diffusion model on a style dataset; evaluate FID/CLIP scores; analyze overfitting.

---

7) SYSTEMS FOR AI (PERFORMANCE & SCALE)

- GPUs/TPUs: memory bandwidth, tensor cores, kernel fusion, streams; NCCL; ZeRO; pipeline/tensor/data parallelism.
- Distributed training: sharding, activation checkpointing, gradient accumulation; mixed precision.
- Inference optimization: graph capture, quantization (INT8/4/NF4), KV-cache, speculative decoding, continuous batching, paged attention.
- Serving: Triton/ONNX/TensorRT; vLLM/TGI; autoscaling; multi-tenant isolation; cost modeling.

Mastery Drills: - Quantize a 7B model to 4-bit; measure throughput/latency vs. baseline across batch sizes and sequence lengths; produce Pareto chart.

---

8) DATA ENGINEERING & GOVERNANCE

- Data lifecycle: sourcing → labeling → QA → versioning → lineage → retention.
- DVC/Lakehouse basics; feature stores; streaming vs. batch; schema evolution.
- Quality: dedupe, PII handling, bias audit, distribution shift detection, data contracts.

Mastery Drills: - Create a data card & model card; implement a drift monitor (PSI/KL) and alert thresholds.

---

9) MLOPS / LLMOPS

- Experiment tracking; reproducibility; model registry; CI/CD for ML; canary/blue-green; rollback.
- Evals: offline regression tests; scenario suites; red-team harnesses; post-deployment telemetry.
- Observability: latency, token usage, cost, quality metrics; SLOs/SLIs; tracing.
- Safety and alignment: jailbreak defenses, content filters, policy enforcement, privacy; attribution/grounding.

Mastery Drills: - Build an evaluation harness with unit tests for prompts, RAG, and tools; wire to CI; fail the build on regression.

---

10) SECURITY, SAFETY, POLICY

- Threats: prompt injection, training data poisoning, model theft, side-channel leaks.
- Controls: isolation, input/output filters, watermarking, rate limiting, provenance (C2PA), canaries.
- Policies: AI risk classifications, human-in-the-loop gates, incident response, audit trails, compliance.

Mastery Drills: - Author a red-team plan; execute a tabletop for data exfiltration via tools; document mitigations.

---

11) PRODUCT, STRATEGY, LEADERSHIP

- Product framing: problem → metric → baseline → intervention → eval → iterate.
- Org patterns: platform vs. applied pods; research vs. production; staffing & IC/EM balance.
- Roadmapping: bets vs. maintenance; kill criteria; tech debt registers; research-to-prod transfer.

Mastery Drills: - Write a one-pager for an AI feature with clear success metrics, telemetry, and experiment plan.

---

12) TOOLS & STACK

- **Languages:** Python; optional C++/Rust for kernels; bash.
- **Frameworks:** PyTorch (primary), JAX (secondary), scikit-learn, Hugging Face stack, LangChain/LlamaIndex (understand, use judiciously).
- **Infra:** Docker, Kubernetes, Ray, Kafka, Airflow; vector DBs (FAISS, Milvus, pgvector, Pinecone); observability (Prometheus/OpenTelemetry).
- **Experimentation:** Weights & Biases/MLflow; A/B platforms.

---

13) CAPSTONE PROJECTS (SELECT 3–5)

1. **Tabular Intelligence:** Production-grade gradient boosting and MLP system with feature store and drift monitor.
2. **From-Scratch GPT-mini:** Minimal language model ($\leq$50M params), trained on small corpus, with perplexity targets and eval suite.
3. **Enterprise RAG:** End-to-end RAG over a realistic corpus; hybrid retrieval; reranker; evals; observability.
4. **Quantized Serving:** INT4 serving with continuous batching; capacity model; autoscaling; SLOs.
5. **RLHF on Toy Task:** Collect preference data; train reward model; run PPO/DPO; safety evals.
6. **Multimodal Agent:** Vision-language agent for document Q&A; tool usage; safety controls.

Deliverables for each: design doc, code, tests, benchmarks, readme, postmortem.

---

14) EVALUATION & CERTIFICATION

- **Written:** math/ML theory; systems & safety; product.
- **Practical:** code review; debug a broken training run; harden an insecure RAG; optimize inference.
- **Oral Defense:** present a capstone; justify tradeoffs; defend against adversarial questioning.

Grading rubric: Accuracy, Rigor, Reproducibility, Security, Cost-Awareness, Communication.

15) 12-WEEK ACCELERATOR (EXECUTION TIMELINE)

W1–2: Foundations — Math refresh; Python/PyTorch; logistic/softmax from scratch; data hygiene. **W3–4: Deep Learning Core** — CNN/RNN/Attention; build mini-Transformer; training stability lab. **W5–6: NLP & RAG** — Tokenization; SFT/LoRA; RAG system v1; evals. **W7–8: Systems** — Distributed training; quantization; serving stack; cost models. **W9–10: MLOps & Safety** — CI for ML, eval harness, red-team scenarios, telemetry. **W11–12: Capstone** — Pick project; deliver with benchmarks and defense.

Expected weekly load: 12–18 focused hours + 1 written assessment + 1 practical.

---

16) YEAR-LONG MASTERY (DEEP OPS)

Quarterly cycles: (Q1) Foundations+DL; (Q2) NLP/RL/Multimodal; (Q3) Systems/Scale; (Q4) Research+Leadership. Publish 2+ public artifacts (papers, talks, open-source) and 1 internal platform.

---

17) CANONICAL RESOURCES (NON-EXHAUSTIVE)

- **Texts:** Goodfellow et al. *Deep Learning*; Murphy *Probabilistic ML*; Bishop *PRML*; Sutton & Barto *RL*; Jurafsky & Martin *Speech & Language Processing*.
- **Courses:** CS231n; CS224n; fast.ai; DeepLearning.AI specializations; Stanford/CMU systems for ML.
- **Papers to Master:** Attention Is All You Need; Scaling Laws; LoRA; QLoRA; RLHF/DPO; RAG triad papers; ViT; CLIP; Diffusion Models.
- **Practice:** Kaggle (tabular & CV), open-source issues, replication studies.

---

18) OPERATIONAL CHECKLISTS

Training Run Readiness: data snapshot, seed control, config file, gradient checks, LR schedule, mixed precision, logging hooks. **Serving Readiness:** load test (p50/p95/p99), max tokens, rate limits, cache sizing, autoscaling, circuit breakers, fallback policy, rollback plan. **RAG Readiness:** ingestion tests, chunking policy, retriever eval, reranker ablation, grounding eval, PI/PHI handling. **Security Readiness:** threat model, jailbreak suite, dependency scan, secrets hygiene, audit logging.

---

19) DAILY DRILLS (20–40 MIN)

- 5–10 min: proof or derivation (rotate topics).
- 10–15 min: code kata (numpy/PyTorch or data plumbing).
- 5–10 min: paper skim + summary.
- 5 min: metrics review of current project.

---

20) INITIAL TASKING (BEGIN TODAY)

1. **Baseline Assessment:**
2. Derive gradient for logistic regression; implement in ~50 lines numpy; verify on toy dataset.
3. Short writeup: bias–variance and regularization tradeoffs (1 page).
4. **Env Setup:** Python 3.11, PyTorch, CUDA toolkit if GPU, Jupyter, MLflow/W&B; repo with `configs/`, `scripts/`, `notebooks/`.
5. **Drill #1:** Implement train/eval loop with early stopping, LR scheduler, gradient clipping; unit tests for loss decreasing.
6. **Read:** Attention Is All You Need (skim math), Goodfellow Ch. 6–8 (optimization), Murphy Ch. 2–4 (probability).

Signal when complete to proceed to Phase 1 live exercises and evals.