

3-Year Detailed Roadmap: Stanford PhD-Level ML + Foundation Models

Intro

Goal: By the end of 3 years, you should be able to:

- Understand ML at PhD rigor
- Train and evaluate foundation models (100M–10B+ scale)
- Design evaluation frameworks & safety metrics
- Create your own PhD-level ML course
- Conduct original research publishable in top conferences

Time Commitment: 35–45 hrs/week

YEAR 1 — Foundations (Weeks 1–52)

Weekly Time Allocation (Approx)

Area	Hours / Week
Math (Linear Algebra + Calc + Prob)	12
Coding + Implementation	8
ML Theory	8
Papers & Notes	5
Deep Dives / Extra	2

MONTHS 1–3 — Math Deep Foundations

Linear Algebra (Vector spaces, SVD, eigen)

Why: You should see models as linear operators.

- **MIT 18.06 Linear Algebra** (Full OCW video + slides):
<https://ocw.mit.edu/courses/18-06-linear-algebra-spring-2010/>
- **Stanford EE263** (slides & notes):
<https://web.stanford.edu/class/ee263/>

Books:

- *Linear Algebra and Learning from Data* — Gilbert Strang
<https://math.mit.edu/~gs/learningfromdata/>

Weekly Tasks

- Derive SVD and interpret PCA in your own words
- Implement SVD from scratch with NumPy

Probability & Statistics

Foundations of Probability & Uncertainty

- **Stanford CS109:**
<https://web.stanford.edu/class/cs109/>
- **MIT 6.041 Probabilistic Systems Analysis:**
<https://ocw.mit.edu/courses/6-041-probabilistic-systems-analysis-and-applied-probability-fall-2010/>

Books

- *All of Statistics* — Wasserman
- *Introduction to Probability* — Blitzstein & Hwang

Papers to Read

- *A Unified View of Gradient Descent* (Sweave basics)
- *What Is a Probability Model?* — Jaynes (philosophical intuition)

Optimization

Courses

- **Convex Optimization** — Stanford EE364a
<https://web.stanford.edu/class/ee364a/>

Books

- *Convex Optimization* — Boyd & Vandenberghe (Free PDF online)

Key Skills

- Understand duality & KKT conditions
- Interpret SGD as approximate Bayesian inference

Machine Learning Theory

Topics to Learn

- Bias / Variance tradeoff
- Regularization
- MLE vs MAP
- Model capacity

Core Texts

- *The Elements of Statistical Learning* — Hastie, Tibshirani, Friedman
<https://web.stanford.edu/~hastie/ElemStatLearn/>
- *Pattern Recognition and Machine Learning* — Bishop

Course

- **Stanford CS229:**
<https://cs229.stanford.edu/>

Coding Mastery (Python + PyTorch)

Courses

- **Python Mastery**
 - Real Python tutorials
- **PyTorch Official Tutorials**
 - <https://pytorch.org/tutorials/>

Projects (have deliverables)

- From-scratch:
 - Linear regression
 - Logistic regression
 - Neural network sigmoid+softmax
- Validate with SciPy/PyTorch

Weekly Paper Reading (5 hrs/week)

Starter Papers:

- *On the Importance of Initialization and Momentum in Deep Learning*
- *Understanding the difficulty of training deep feedforward networks*
- *A Brief Survey of Deep Learning*

Create annotated notes as mini blog posts.

End of Year 1 Deliverables

- ✓ Math notes (organized PDF / folder)
- ✓ Working Python + PyTorch toolchain
- ✓ Mini-ML models with tests
- ✓ Portfolio README documenting all experiments

🎯 YEAR 2 — Deep Learning, Transformers, Systems

Weekly Time Allocation

Area	Hours / Week
Deep Learning Theory	10
Transformer & NLP	8

Systems + Scale	8
Implementation Projects	12
Reading + Summaries	5

Deep Learning Internals

Courses

- **Stanford CS231n (ConvNets & DL)**
<https://cs231n.stanford.edu/>
- **Stanford CS230**
<https://cs230.stanford.edu/>

Book

- *Deep Learning* — Goodfellow, Bengio, Courville
<https://www.deeplearningbook.org/>

Focus Areas

- Backprop algebra
- Initialization
- BatchNorm / LayerNorm
- Loss surfaces

Weekly Tasks

- Re-derive gradients manually
- Implement a tiny autograd engine

Transformers & Sequences

Courses

- **Stanford CS25 — Transformers**
<https://web.stanford.edu/class/cs25/>
- **Stanford CS224n — NLP with Deep Learning**
<https://web.stanford.edu/class/cs224n/>

Core Papers

- *Attention Is All You Need*
<https://arxiv.org/abs/1706.03762>
- *BERT: Pre-training of Deep Bidirectional Transformers*
<https://arxiv.org/abs/1810.04805>
- *Transformer-XL: Attentive Language Models Beyond a Fixed Length*
<https://arxiv.org/abs/1901.02860>

Projects

- Train a 100M-parameter transformer
- Tokenizer from scratch (BPE/Unigram)

Large Model Systems

Courses

- **Stanford CS336 — Language Models from Scratch**
<https://cs336.stanford.edu/>
- **CMU 10-414 / 10-714 — Deep Learning Systems**
<https://dlsyscourse.org/>

Topics


- GPU memory management
- Data-parallel + model-parallel
- Checkpointing
- Autotuning


Implementations (Core Projects)


Deliverables

- Training pipeline (config + checkpoints)
- Logging & reproducibility
- Hyperparameter sweep system
- Experiments with learning-rate schedules (AdamW, CosineLR)

Year 2 Papers to Read (Deeply)

 *Scaling Laws for Neural Language Models*
<https://arxiv.org/abs/2001.08361>

 *Chinchilla: Training Compute-Optimal Large Models*
<https://arxiv.org/abs/2203.15556>

 *On Layer Normalization in the Transformer Architecture*
<https://arxiv.org/abs/2002.04745>

End of Year 2 Deliverables

- ✓ Trainable transformer pipeline
- ✓ Deep learning research notebook
- ✓ Systems experimentation logs
- ✓ Clear writeups (Markdown / blog)

YEAR 3 — Foundation Models, Evaluation, Alignment, Research

Weekly Time Allocation

Area	Hours/Week
Foundation Models Theory	10
Evaluation Frameworks	10

Alignment & Safety	10
Research Projects	10
Writing & Teaching	5

LLM Evaluation & Benchmarks

Course:

- **Stanford CS324 — Large Language Models**
<https://stanford-cs324.github.io/>

Papers

- *InstructGPT* — Fine-tuning LMs using RLHF
<https://arxiv.org/abs/2203.02155>
- *Red Teaming Language Models* (Anthropic)
<https://arxiv.org/abs/2202.03286>
- *Beyond Accuracy: Behavioral Testing of NLP Models*
<https://arxiv.org/abs/1804.08452>

Projects

- Build an eval harness
 - Perplexity
 - Robustness
 - Safety metrics

Alignment & Safety

Core Papers

- *DPO: Direct Preference Optimization*
<https://arxiv.org/abs/2309.02112>
- *Constitutional AI*
<https://arxiv.org/abs/2212.08073>

- *Measuring Moral Foundations in LLMs*
<https://arxiv.org/abs/2307.02638>

Focus

- Preference modeling
- Tradeoffs: harmless vs helpful
- Red-teaming loops

Research Roadmap

1. Replication

Pick a recent foundation model paper and replicate:

- Training pipeline
- Evaluation
- Ablations

Examples

- *LLaMA*
<https://arxiv.org/abs/2302.13971>
- *GPT-NeoX*
<https://arxiv.org/abs/2204.06745>

2. Original Research

Choose a gap from replication and explore:

- Efficiency improvements
- Safety evaluation improvements
- New eval metrics

Write & submit to:

- NeurIPS / ICLR workshops
- ArXiv

Teaching + Curriculum Creation

Create:

- 12-week PhD-level course syllabus
- Assignments + autograders
- Reading list + rubric

Make it public to build credibility.

Master List of Papers With Links

Topic	Paper	Link
Transformers	<i>Attention Is All You Need</i>	https://arxiv.org/abs/1706.03762
Scaling	<i>Scaling Laws...</i>	https://arxiv.org/abs/2001.08361
Compute-Optimal	<i>Chinchilla</i>	https://arxiv.org/abs/2203.15556
Instruction Tuning	<i>InstructGPT</i>	https://arxiv.org/abs/2203.02155
Alignment	<i>DPO</i>	https://arxiv.org/abs/2309.02112
Safety	<i>Red Teaming LMs</i>	https://arxiv.org/abs/2202.03286
... (you should build a 150+ paper list as a spreadsheet)		

Tools to Use

- **Weights & Biases / MLflow** — experiment tracking
- **GitHub Actions** — CI for training pipelines

- **Docker** — reproducibility
- **T5X / Megatron / ColossalAI** — large-model training frameworks