The Physics of Learning: From Spin Glass to Neural Fields

A Complete Research Project Structure

Research Project Guide
Version 1.0
December 30, 2025

Abstract: This comprehensive research project explores the deep connections between statistical physics and artificial intelligence. Starting from the Ising model of magnetism, we progress through Hopfield networks, Boltzmann machines, and culminate in the study of Neural Network Gaussian Processes and their connection to quantum field theory. This project is designed as a 12-17 week structured investigation suitable for advanced undergraduate or graduate research, with potential for publication-quality results.

Table of Contents

Phase 0: Theoretical Foundation (1 week)

Build the mathematical bedrock

Before diving into implementation, you need to establish a solid theoretical foundation. This phase focuses on understanding the mathematical and physical principles that underpin the entire project.

Deliverables:

Literature Review Document covering:

• Statistical mechanics basics (partition function, Boltzmann distribution)

• Energy minimization in physical systems

• The Ising model mathematical formulation

• Hopfield (1982) and Hinton's key papers

• Neural Tangent Kernel theory (Jacot et al. 2018)

> ***Why This Matters:*** *You need the language before the experiments. This prevents 'black box' syndrome later and ensures you understand the theoretical connections you'll be demonstrating.*

Phase 1: The Ising Model - Emergence of Order (2-3 weeks)

Simulate physics, understand phase transitions

1.1 Core Implementation

The Ising model is a mathematical model of ferromagnetism in statistical mechanics. You'll implement a 2D lattice where each site has a spin that can point up (+1) or down (-1).

Key components:

```
# Key components: - 2D lattice (start with 20×20, scale to 100×100) -
Metropolis-Hastings algorithm - Energy calculation: H = -J Σ σ■σ■ -
Temperature scheduling
```

1.2 Experiments to Run

Critical Temperature Discovery

• Sweep T from 0.5 to 5.0

• Plot: Magnetization vs Temperature

• Identify Curie temperature (≈2.27 for 2D Ising)

Equilibration Dynamics

• Start from random configuration

• Measure: steps to equilibrium at different T

• Create time-lapse visualization

Correlation Length

• Measure: How far does one spin's influence reach?

• Plot: Correlation vs distance at different temperatures

1.3 Deliverables

• **Animated visualization** of spin evolution

• **Phase diagram** showing order/disorder transition

• **Key insight:** "Collective behavior emerges from local rules"

Phase 2: Hopfield Networks - Memory as Energy Minima (2-3 weeks)

Repurpose physics for computation

Now you'll transform the Ising model into a computational tool. Instead of physical spins, you'll have artificial neurons that can store and recall patterns.

2.1 Core Implementation

```
# Hebbian learning rule: w_ij = (1/N) Σ_μ ξ■^μ ξ■^μ Where: - ξ^μ = stored pattern μ - N = number of neurons
```

2.2 Experiments to Run

1. Single Pattern Storage

• Store 'Y' (for Yehia) in 10×10 grid

• Test with: 10%, 25%, 50% corruption

• Visualize: corruption → recovery process

• Measure: convergence steps

2. Capacity Limits

• Store 1, 2, 3, 5 patterns (letters: Y, E, H, I, A)

• Test recall accuracy

• Find capacity limit (~0.138N as theory predicts)

• Document: spurious states (false memories)

3. Energy Landscape Visualization

• For 2-3 stored patterns

• Use PCA to project high-D space to 2D

• Show: energy basins around stored patterns

2.3 Critical Analysis

Compare to Ising Model: *Same math, different meaning*

- **Ising:** J is physics (material property)
- **Hopfield:** w is learned (from data)
- **Key insight:** "Memory is a stable energy configuration"

2.4 Deliverables

- **Interactive demo:** Upload corrupted pattern, watch recovery
- **Capacity analysis:** Theory vs. experiment plots
- **Energy landscape:** 3D visualization

Phase 2.5: The Missing Link - Boltzmann Machines (1-2 weeks)

Bridge discrete and continuous, add hidden layers

2.5.1 Why This Phase?

Hopfield networks have no hidden units. Real neural networks do. This phase bridges the gap between simple associative memory and deep learning architectures.

2.5.2 Implementation

Restricted Boltzmann Machine (RBM)

• Visible layer (10 units) + Hidden layer (5 units)

• Learn to encode simple patterns

• Use Contrastive Divergence for training

2.5.3 Key Experiment

• Train RBM on hand-drawn digits (simplified MNIST: just 0, 1, 2)

• Show: Hidden units learn features automatically

• Visualize: What each hidden unit 'detects'

2.5.4 Deliverable

• **Demonstration:** RBM generates new 'digit-like' patterns
• **Key insight:** "Hidden layers learn distributed representations"

Phase 3: Neural Network Gaussian Processes (3-4 weeks)

The infinite width limit and field theory

3.1 Theoretical Setup

Study the limit: What happens as width $\rightarrow \infty$?

> ***Key Mathematical Result (Neal, 1995):***
> *For a neural network $f(x; \theta)$ with random weights $\theta \sim N(0, \sigma^2)$:*
>
> *As width $\rightarrow \infty$: $f(x) \sim GP(0, K(x,x'))$*
>
> *Where K is the Neural Network Gaussian Process (NNGP) kernel*

3.2 Implementation Path

3.2.1 Finite Width Networks

```
# Build networks of increasing width: widths = [10, 50, 100, 500, 1000, 5000]
For each width: - Initialize random weights - Compute outputs on test points
- Calculate empirical mean and covariance
```

3.2.2 Analytical GP Kernel

```
# For 1-layer network with ReLU: K(x, x') = (σ_w²/π) * ||x|| ||x'|| * (sin(θ)
+ (π - θ)cos(θ)) Where θ = arccos(x·x' / (||x|| ||x'||))
```

3.2.3 Convergence Experiment

- **Dataset:** 1D regression (y = sin(x) + noise)
- **Compare:** Finite width NN predictions (ensemble of 100 networks) vs. GP with NNGP kernel (exact)
- **Plot:** Predictions + uncertainty bands
- **Measure:** KL divergence between distributions as width increases

3.3 Advanced: The Field Theory Connection

3.3.1 Correlation Functions

```
In physics: C(x, x') = ■ϕ(x)ϕ(x')■
In ML: K(x, x') = E[f(x)f(x')]
```

Experiment:

- Compute 2-point correlation functions for your networks
- Show they match covariance structure of GP

- **Visualization:** Heat map of K(x, x') for 2D input space

3.3.2 Feynman Diagram Interpretation (Optional but impressive)

- Show how network computation graphs → Feynman diagrams
- Each layer = propagator
- Each nonlinearity = interaction vertex
- Calculate 'loop corrections' (finite width effects)

3.4 Deliverables

- **Convergence plot:** Showing finite → infinite width
- **Regression comparison:** NN vs GP predictions
- **Kernel visualization:** Heat maps of learned similarity
- **Research notebook:** Detailed mathematical derivations

Phase 4: Synthesis & Novel Contribution (2-3 weeks)

Your original research contribution

4.1 Comparative Analysis Across All Phases

Create a unified framework showing how all systems relate:

System	Energy Function	Update Rule	Equilibrium	Learning
Ising	H = -J Σ σ■σ■	Metropolis	Aligned spins	N/A
Hopfield	H = -Σ w■■σ■σ■	Deterministic	Stored pattern	Hebbian
RBM	H = -Σ w■■v■h■	Gibbs sampling	Data distribution	Gradient
NNGP	KL[q■p]	Gradient descent	Function space	Infinite width

4.2 Novel Experiments (Choose 1-2)

Option A: Finite Width Corrections

• Measure deviation from GP predictions

• Connect to 'interaction strength' in field theory

• Plot: How many 'loops' (layers) needed for quantum corrections?

Option B: Phase Transitions in Learning

• Train networks at different widths

• Identify critical width where behavior changes

• Draw analogy to Ising phase transition

Option C: Transfer Learning as Symmetry

• Store patterns in Hopfield network

• Fine-tune with new pattern

• Measure: 'Interference' between old and new memories

• Connect to: Symmetry breaking in physics

4.3 Deliverable

- **Original research finding** (even if small)
- **Publication-ready figures**
- **Clear thesis:** "We show that [X] in neural networks corresponds to [Y] in statistical physics"

Phase 5: Communication (1-2 weeks)

Package your work for impact

5.1 Written Outputs

1. Research Paper (8-12 pages)

• Abstract

• Introduction (motivation + physics-ML connection)

• Methods (each phase)

• Results (with figures)

• Discussion (limitations + future work)

• Conclusion

2. GitHub Repository

• Clean, documented code

• README with setup instructions

• Jupyter notebooks for each phase

• Requirements.txt

3. Blog Post (Medium/personal site)

• Accessible version for general audience

• Interactive visualizations (plotly/d3.js)

5.2 Visual Outputs

1. **Poster** (for conferences/career fair)
2. **5-minute video** (screencast of key results)
3. **Interactive demo** (streamlit or gradio app)

Timeline (12-17 weeks total)

Week	Phase	Focus
1	Phase 0	Theoretical Foundation
2-4	Phase 1	Ising Model Implementation
5-7	Phase 2	Hopfield Networks
8-9	Phase 2.5	Boltzmann Machines
10-13	Phase 3	NNGP & Field Theory
14-15	Phase 4	Novel Research Contribution
16-17	Phase 5	Communication & Publication

Success Criteria

Minimum Viable Project:

✓ Working Ising simulation with phase transition

✓ Hopfield network that recovers corrupted patterns

✓ Demonstration of finite → infinite width convergence

Strong Project:

✓ Above + RBM implementation

✓ Quantitative comparison of correlation functions

✓ Clean, reproducible code

✓ Well-written paper

Exceptional Project:

✓ Above + novel insight/experiment

✓ Publication-quality figures

✓ Conference paper submission (e.g., NeurIPS workshop)

✓ Open-source tool others can use

Key Resources

Essential Papers

1. **Hopfield (1982)** - "Neural networks and physical systems with emergent collective computational abilities"

2. **Neal (1995)** - "Bayesian Learning for Neural Networks" (PhD Thesis)

3. **Lee et al. (2018)** - "Deep Neural Networks as Gaussian Processes" (ICLR)

4. **Jacot et al. (2018)** - "Neural Tangent Kernel: Convergence and Generalization in Neural Networks" (NeurIPS)

5. **Matthews et al. (2018)** - "Gaussian Process Behaviour in Wide Deep Neural Networks" (ICLR)

Code Libraries & Tools

• **NumPy/SciPy** - Core numerical computation

• **scikit-learn** - GP regression implementation

• **JAX** - Automatic differentiation for neural networks

• **Neural Tangents** library (Google) - NNGP kernels

• **PyTorch** or **TensorFlow** - Deep learning frameworks

• **matplotlib** + **seaborn** - Static visualizations

• **plotly** - Interactive visualizations

• **manim** - Animation library (optional, for video explanations)

Additional Learning Resources

• **Statistical Mechanics:** Kardar - "Statistical Physics of Particles"

• **Neural Networks:** Goodfellow et al. - "Deep Learning" (Chapter 20: Deep Generative Models)

• **Gaussian Processes:** Rasmussen & Williams - "Gaussian Processes for Machine Learning"

• **Quantum Field Theory:** Peskin & Schroeder (for the ambitious!)

Why This Structure Works

1. Builds intuition progressively

Physics → Computation → Theory. Each phase builds naturally on the previous one, creating a coherent narrative arc.

2. Each phase stands alone

You can stop at any phase and still have a complete, presentable project. This provides natural checkpoints for evaluation.

3. Multiple difficulty levels

Choose depth based on time and interest. The minimum viable project is achievable, while the exceptional project challenges advanced students.

4. Portfolio-ready

Demonstrates physics knowledge, machine learning understanding, mathematical rigor, and coding proficiency - all highly valued skills.

5. Research-grade potential

Phase 4 is specifically designed to enable novel contributions that could lead to actual publications or conference presentations.

6. Interdisciplinary appeal

Bridges multiple fields (physics, computer science, mathematics, neuroscience), making it attractive to diverse audiences and reviewers.

> ***Final Thoughts:*** *This project represents a genuine opportunity to contribute to our understanding of the deep connections between physics and artificial intelligence. By following this structured approach, you'll not only learn about these connections but also develop the skills and produce the artifacts necessary for a strong research portfolio. Remember: the goal is not just to replicate known results, but to develop your own intuition and potentially discover new insights along the way.*

Good luck with your research journey!