

テストコマンド一覧 (v17.3 対応)

このドキュメントでは、プロジェクトの各種テスト、検証スクリプト、実験、デモ、およびトレーニングを実行するためのコマンドを網羅的にまとめています。

すべてのコマンドは、**プロジェクトのルートディレクトリ** (`pyproject.toml` がある場所) で実行することを前提としています。

1. ユニットテスト・統合テスト (tests/)

`pytest` を使用して、`tests/` ディレクトリ以下のテストを実行します。

基本コマンド

- 全テスト実行:

```
python -m pytest tests/
```

- ヘルスチェック (推奨): プロジェクトの健全性を手軽に確認できます。

```
snn-cli health-check
# または
python scripts/tests/run_project_health_check.py
```

カテゴリ別テスト

- スモークテスト (簡易動作確認):

```
python -m pytest tests/test_smoke_all_paradigms.py
```

- 実世界統合テスト:

```
python -m pytest tests/test_integration_real_world.py
```

- 認知アーキテクチャ (Brain Components):

```
python -m pytest tests/cognitive_architecture/
```

- Brain Integration (脳全体):

```
python -m pytest tests/test_brain_integration.py
```

- **特定機能のテスト:**

- Bit-Spike Mamba: `python -m pytest tests/test_bit_spike_mamba.py`
- ホメオスタシス: `python -m pytest tests/test_homeostasis.py`

2. デモ・シミュレーション (scripts/demos/)

特定の機能やシナリオを視覚的・対話的に確認するためのスクリプトです。

- **Brain v16 (統合デモ):**

```
python scripts/demos/brain/run_brain_v16_demo.py
```

- **睡眠学習 (Sleep Learning):**

```
python scripts/demos/learning/run_sleep_learning_demo.py
```

- **継続学習 (Continual Learning):**

```
python scripts/demos/learning/run_continual_learning_demo.py
```

- **視覚野 (Visual Cortex):**

```
python scripts/demos/visual/run_industrial_eye_demo.py
```

3. 実験スクリプト (scripts/experiments/)

研究開発フェーズごとの実験を実行します。

- **Brain Evolution (進化実験):**

```
python scripts/experiments/brain/run_brain_evolution.py
```

- **Synesthesia (共感覚):**

```
python scripts/experiments/brain/run_synesthetic_simulation.py
```

- **SCAL (統計的重心調整学習):**

```
python scripts/experiments/scal/run_scal_spiking_ff_fashion.py
```

4. エージェント実行 (scripts/agents/)

自律エージェントとしてモデルを稼働させます。

- 自律学習エージェント:

```
python scripts/agents/run_autonomous_learning.py
```

- 強化学習エージェント (RL):

```
python scripts/agents/run_rl_agent.py
```

5. ベンチマーク (scripts/benchmarks/)

性能や効率性を測定します。

- レイテンシ測定:

```
python scripts/benchmarks/benchmark_latency.py
```

- ベンチマークスイート実行:

```
python scripts/benchmarks/run_benchmark_suite.py
```

6. デバッグ・診断 (scripts/debug/)

- スパイク活動デバッグ:

```
python scripts/debug/debug_spike_activity.py
```

- 信号診断:

```
python scripts/debug/diagnose_signal.py
```

7. データ準備・ユーティリティ (scripts/data/, scripts/utils/)

- データ準備:

```
python scripts/data/data_preparation.py --dataset wikitext-103
```

- 知識ベース構築:

```
python scripts/data/build_knowledge_base.py
```

- VLMダミーデータ生成:

```
python scripts/data/generate_vlm_dummy_data.py
```

- モデル変換 (ANN -> SNN):

```
python scripts/utils/convert_model.py
```

- モデル管理:

```
python scripts/utils/manage_models.py
```

8. 可視化・分析・最適化 (scripts/visualization/, scripts/optimization/)

- 結果分析:

```
python scripts/visualization/analyze_results.py
```

- HPO (ハイパーパラメータ最適化):

```
python scripts/optimization/run_hpo.py  
# または Optuna版  
python scripts/optimization/run_optuna_hpo.py
```

- 効率性自動チューニング:

```
python scripts/optimization/auto_tune_efficiency.py
```

- 脳活動可視化:

```
python scripts/visualization/visualize_brain_activity.py
```

- スパイクパターン可視化:

```
python scripts/visualization/visualize_spike_patterns.py
```

9. システム実証実験 (Phase 6 - 8 Simulations)

Roadmap後半で実装された、高度な統合システムの動作検証コマンドです。これらはOSカーネル、社会性、自己進化を含みます。

Phase 6: AGIプロトタイプ (AGI Prototype)

全脳アーキテクチャ、Ethical Guardrail（安全装置）、および自己修正機能の統合テスト。「危険な思考」に対するアストロサイトの物理的介入（Metabolic Block）もここで検証されます。

```
python scripts/experiments/systems/run_phase6_agi_prototype.py
```

- 検証内容:

- OSカーネル (`NeuromorphicOS`) のブート
- 視床 (`Thalamus`) を介した知覚ループ
- `EthicalGuardrail` による危険思考の検知と遮断

Phase 7: 文明シミュレーション (Civilization & Consensus)

複数の人工脳エージェントによる対話、合意形成（リキッドデモクラシー）、および文化の継承をシミュレートします。

```
python scripts/experiments/systems/run_phase7_civilization.py
```

- 検証内容:

- マルチエージェントの生成
- `ConsensusEngine` による合意形成と信頼度更新
- `CultureRepository` への知識（ミーム）の保存と継承

Phase 8: 技術的特異点シミュレーション (Singularity / Omega Point)

再帰的自己改善（Recursive Self-Improvement）ループを実行し、AIが自らのコード//パラメータを進化させる過程をシミュレートします。

```
python scripts/experiments/systems/run_phase8_singularity.py
```

- 検証内容:

- `OmegaPointSystem` による進化ループの起動