

Neuromorphic Research OS: 学習・観測ガイド (v6.0)

本プロジェクトにおける「学習 (Training)」は、単なるパラメータ最適化ではなく、**「経験に伴う脳構造の自己組織化」と定義されます。また、「推論 (Inference)」は静的な計算ではなく、「リアルタイムの神経活動現象」**として扱われます。

このドキュメントでは、Neuromorphic OS上での学習プロセスの回し方と、その観測方法について解説します。

1. OS上での学習サイクル (Structural Learning)

Neuromorphic OS v6.0以降では、以下のプロセスで学習が進行します。このプロセスは自動化されており、ユーザーは環境（入力データや報酬）を提供するだけです。

学習のメカニズム

1. 覚醒 (Wake):

- Forward-Forward則 / STDP:** 局所的な可塑性によりシナプス強度が変化。
- ドーパミン (Dopamine):** 正解や報酬により可塑性が一時的に強化される（強化学習）。
- エピソード記録:** 重要な入力パターンが海馬バッファに一時保存される。

2. 睡眠 (Sleep):

- 海馬リプレイ (Replay):** 覚醒時のパターンが高速再生（夢）され、記憶が皮質へ転送される。
- シナプス恒常性 (Pruning):** 不要・微弱なシナプスが物理的に削除される（忘却による整理）。
- シナプス生成 (Synaptogenesis):** 新しい接続がランダムに芽生え、新たな学習の準備をする。

実行コマンド

```
# 記憶定着と構造変化の実験（推奨）
python scripts/experiments/learning/run_memory_consolidation.py
```

パラメータ調整

学習の挙動を変えたい場合は、スクリプト内の `config` や `os_kernel` の設定を変更します。

- `max_energy`: 覚醒時間の長さを制御。
- `os_kernel.reward(amount=...)`: ドーパミンの放出量を調整。

2. 観測と分析 (Observation)

学習の結果は「正解率」という数値だけでなく、**脳の状態変化**として評価します。

主要な観測指標

- **Synapse Count (Syn):** 脳内の有効な接続総数。睡眠中に減少し（刈り込み）、覚醒後期に増加する（生成）ダイナミクスが健全です。
- **Energy / Fatigue:** 代謝のリズムが保たれているか。
- **Consciousness Level:** Global Workspaceへの情報統合強度。

可視化

実験後に生成されるJSONデータをグラフ化します。

```
python scripts/visualization/plot_memory_learning.py
```

成功のサイン: グラフ上で「シナプス数がV字回復している（減ってから増える）」かつ「その後に精度（Accuracy）が向上している」場合、構造的可塑性が正しく機能しています。

3. 単体モデルの学習 (Component Training)

OSカーネル全体ではなく、特定のSNNモデル（Spiking CNNなど）単体の性能を評価したい場合に使用する、従来型の学習スクリプトです。これらはPyTorch標準の学習ループに近い形式で動作します。

A. 高精度レシピ (Recipes)

`snn_research/recipes/` 以下のスクリプトは、特定のタスク（MNIST, CIFAR-10）に特化したチューニング済みモデルを学習させます。

- **MNIST 学習:**

```
python -c "from snn_research.recipes.mnist import run_mnist_training; run_mnist_training"
```

- **CIFAR-10 学習:**

```
python -c "from snn_research.recipes.cifar10 import run_cifar10_training; run_cifar10_training"
```

B. 汎用トレーナー

設定ファイルを指定して任意のモデルを学習させます。

```
PYTHONPATH=. python scripts/training/train.py --model_config configs/models/small
```

4. Webインターフェース (App Demo)

学習した脳の挙動をブラウザ上でインタラクティブに確認するためのWebサーバーです。

- **起動コマンド:**

```
python app/main.py
```

起動後、ブラウザで `http://localhost:8000` にアクセスします。

- **統合デモ (Unified Perception):** 視覚・言語・運動野の連携デモを行います。

```
python app/unified_perception_demo.py
```