

SNNプロジェクト: 学習・推論コマンドガイド (v17.3)

このドキュメントでは、モデルの学習（Training）と推論・デモ（Inference/Demo）を実行するための主要なコマンドについて解説します。プロジェクトディレクトリのルートで実行してください。

⚠ 実行前の注意

スクリプトを実行する際、モジュールのインポートエラー（`ModuleNotFoundError`）が発生する場合は、環境変数 `PYTHONPATH` にカレントディレクトリを追加してください。

```
# Mac/Linux
export PYTHONPATH=.

# Windows (PowerShell)
$env:PYTHONPATH="."
```

または、各コマンドの先頭に `PYTHONPATH=.` を付けて実行します (例: `PYTHONPATH=. python scripts/...`)。

1. 学習 (Training)

A. 高精度レシピ (High-Performance Recipes)

特定のタスク（CIFAR-10, MNIST）で最高精度を目指すための推奨スクリプトです。これらは `snn_research/recipes/` 内に定義されており、以下のように実行します。

- **CIFAR-10 学習 (96% Aim)**

```
# Pythonモジュールとして実行
python -c "from snn_research.recipes.cifar10 import run_cifar10_training; run_cifar10_training"
```

- **MNIST 学習**

```
# Pythonモジュールとして実行
python -c "from snn_research.recipes.mnist import run_mnist_training; run_mnist_training"
```

B. 汎用トレーナー (Generic Trainer)

設定ファイル（`configs/`）を指定して、様々なアーキテクチャのモデルを学習させます。

```
# Spiking CNNの設定で学習
PYTHONPATH=. python scripts/training/train.py --config configs/experiments/cifar10

# デフォルト設定（小規模モデル）で学習
```

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

注意: 現在、データセット自動ダウンロード機能の一部（WikiTextなど）はモジュール構成の変更により無効化されています。`smoke_test_data.jsonl` は自動生成されます。

C. 特定モデルの学習スクリプト

- **Spiking VLM (Vision-Language Model) 学習:**

```
python scripts/training/train_spiking_vlm.py
```

- **Planner (推論エンジン) 学習:**

```
python scripts/training/train_planner.py
```

- **SCAL (Statistical Centroid Alignment Learning):** 勾配計算を行わない高速学習手法です。

```
python scripts/training/run_improved_scal_training.py \  
--config configs/templates/base_config.yaml \  
--model_config configs/models/small.yaml
```

2. 推論・デモ (Inference & Demo)

Webアプリ/APIサーバー

FastAPIサーバーを起動し、ブラウザから対話や画像認識を行います。

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

統合デモ (Unified Perception)

視覚・言語・運動野を統合したデモを実行します。

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

CLIによる推論

`snn-cli` コマンドがインストールされている場合、以下のように使用できます。

```
# ヘルスチェック  
snn-cli health-check  
  
# レシピ一覧（実装されている場合）
```

```
# snn-cli recipe list
```

もし `snn-cli` コマンドがパスに通っていない場合は、以下のように実行します。

```
python snn-cli.py health-check
```

3. 高度な学習パラダイム

生物学的妥当性を重視した学習手法です。

- **STDP (Spike-Timing Dependent Plasticity):**

```
python scripts/experiments/learning/run_stdp_learning.py
```

- **オンチップ学習 (On-Chip Learning):** エッジデバイス上での適応を想定した学習です。

```
python scripts/experiments/learning/run_on_chip_learning.py
```

- **蒸留学習 (Distillation):**

```
python scripts/experiments/learning/run_distillation_experiment.py
```