Hyperspectral Data Processing Pipeline Documentation

Preprocessing of Large-Scale Hyperspectral Data for Compatibility with Convolutional Neural Networks (CNNs) and Other Machine Learning Algorithms

Parth Chopra
Dr. Ali Hatef
COSC-4897 – Honours Research II

Nipissing University

Table of Contents

Hyperspectral Data Processing Pipeline Documentation

Overview

This pipeline processes hyperspectral data through a series of steps: baseline correction, normalization, dimensionality reduction (PCA), and conversion to HDF5 format. It is designed for large hyperspectral cubes and includes utilities for handling `NaN` values and memory efficiency.

Dependencies

```
Python 3.7+
```

Required libraries:

```
pip install numpy scipy scikit-learn h5py spectral
```

Configuration

config.py

Sets file paths and directories. Modify `base_dir` to match your dataset location:

```
base_dir = "datasets"  # Update this path
```

Key Paths

- `data_path`: Input hyperspectral cube (.npy format).
- `baseline_path`: Baseline anchor wavelengths (.txt file).
- `envi_header_path`: ENVI header file (.hdr).
- `output_pca_path`: Output path for PCA-transformed data.
- `final_output_path`: Reshaped cube path (used for HDF5 conversion).

Modules

1. main.py

Workflow:

1. **Load Data**: Reads hyperspectral cube, baseline, and ENVI header.
2. **Baseline Correction**: Subtracts interpolated baseline from spectra.
3. **Normalization**: Normalizes cube by absorbance at a target wavenumber (default: 1650 cm^{-1}).
4. **PCA**: Reduces dimensionality using Incremental PCA (16 components by default).
5. **HDF5 Conversion**: Saves the final cube as an HDF5 file.

2. load_utils.py

Functions:

- `load_envi_header(header_path)`: Loads ENVI header metadata.
- `load_cube(path)`: Loads hyperspectral cube as a memory-mapped NumPy array (efficient for large files).
- `load_baseline(path)`: Loads baseline anchor wavelengths from a text file.

3. preprocess.py

Function: `baseline_correction(cube, wavelengths, anchor_wavelengths)`

- **Inputs**:
  - `cube`: Hyperspectral cube (shape: [rows, cols, bands]).
  - `wavelengths`: Array of all band wavelengths.
  - `anchor_wavelengths`: Baseline anchor wavelengths.
- **Process**: Linear interpolation of baseline values and subtraction from spectra.
- **Output**: Baseline-corrected cube (same shape as input).

4. normalize.py

- *Function*: `normalize_hyperspectral_cube(cube, wavenumbers, target_wavenumber=1650.0)`
- **Inputs**:
    - `cube`: Baseline-corrected cube.
    - `wavenumbers`: Array of wavenumbers (size: bands).
    - `target_wavenumber`: Wavenumber for normalization (default: 1650 cm^{-1}).
- **Process**: Divides each spectrum by the absorbance at the target wavenumber.
- **Output**: Normalized cube (same shape as input).

5. pca_utils.py

- *Function*: `incremental_pca_npy (npy_file, output_file, n_components=16, batch_size=5000)`
- **Inputs**:
    - `npy_file`: Normalized hyperspectral cube (NumPy array).
    - `output_file`: Path to save PCA-transformed cube.
    - `n_components`: Number of principal components (default: 16).
    - `batch_size`: Pixels processed per batch (adjust for memory constraints).
- **Process**:
    - Reshapes cube to 2D and imputes NaN values using mean.
    - Fits Incremental PCA in batches.
    - Transforms data and reshapes to `[rows, cols, n_components]`.
- **Output**: .`npy` file with PCA-transformed cube.

6. convert.py

Function: `convert_npy_to_h5(npy_file,h5_file,dataset_name="ftir_dataset")`

- Converts a NumPy array (.npy) to HDF5 format.
- **Inputs**:
    - npy_file: Path to input .npy file.
    - h5_file: Output HDF5 filename.
    - dataset_name: Name of the HDF5 dataset (default: "ftir_dataset").

- **Output:**

Usage

1. **Configure Paths**: Update config.py with your dataset paths.
2. **Run Pipeline**:
   ```
   Python main.py
   ```
3. **Outputs**:
   - `pca_cube.npy`: PCA-transformed data.
   - `output.h5`: Final HDF5 file.

Input/Output Formats:

- **Input Cube**: 3D NumPy array `([rows, cols, bands]).`
- **Baseline File**: Text file with one wavelength per line.
- **ENVI Header**: Standard `.hdr` file with wavelength metadata.
- **Outputs**:
  - `.npy` for intermediate results.
  - `.h5 (HDF5)` for final storage.

References:

Berisha, Sebastian, et al. "Deep Learning for FTIR Histology: Leveraging Spatial and Spectral Features with Convolutional Neural Networks." *Analyst . 2019*, vol. 144, no. 5, 1 Jan. 2019, pp. 1642–1653, https://doi.org/10.1039/c8an01495g. Accessed 21 July 2023.

"SIproc: An Open-Source Biomedical Data Processing Platform for Large Hyperspectral Images." *The Analyst*, vol. 142, no. 8, 23 Nov. 2016, pp. 1350–1357, https://doi.org/10.1039/c6an02082h. Accessed 28 Apr. 2025.