Optilearn.nn.EarlyStopping

08 November 2024 13:12

EarlyStopping

The EarlyStopping class is a callback utility designed to halt training of a model when a specified metric ceases to improve. It monitors metrics such as accuracy or loss, based on parameters specified by the user, and can stop training if no improvement is observed within a defined patience period. It also optionally restores the model's best weights from a checkpoint.

Purpose

This class monitors a metric during training (e.g., val_accuracy, val_loss). When a model's performance stops improving (defined by the patience and min_delta parameters), training halts, preserving resources and potentially preventing overfitting.

Parameters

- **monitor** (str, optional): Metric to observe for early stopping. Accepted values are 'accuracy', 'val_accuracy', 'loss', or 'val_loss'. Default is 'val_accuracy'.
- **mode** (str, optional): Defines whether an increase ('max') or decrease ('min') in the monitored metric is considered an improvement. If set to 'auto', the class will automatically infer the appropriate mode based on the metric.
- **patience** (int, optional): Number of epochs with no improvement after which training will be stopped. Default is 5.
- **verbose** (int, optional): Controls verbosity. Set to 1 to enable progress messages when early stopping is triggered. Default is 0.
- **min_delta** (float, optional): Minimum change in the monitored metric to qualify as an improvement. Default is 0.
- **baseline** (float, optional): Reference value for the metric. Training halts if the monitored metric meets or exceeds (or falls below, depending on mode) this value. Default is 1.0.
- **restore_best_weights** (bool, optional): If True, the model will revert to weights from the epoch with the best recorded metric. Default is False.

Methods

__init__(self, monitor='val_accuracy', mode='auto', patience=5, verbose=0, min_delta=0, baseline=1.0, restore_best_weights=False)

Initializes the EarlyStopping instance with the specified monitoring parameters.

check(self, metrics_values, number_of_epoch, pre_value, time_step)

Evaluates whether training should halt based on the current value of the monitored metric. If the metric fails to improve by at least min_delta within patience epochs, training stops.

Parameters:

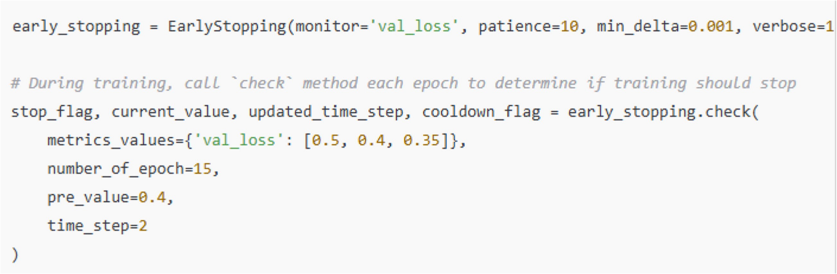
- **metrics_values** (dict): Dictionary with metric values for each epoch, where the monitored metric is a key.
- **number_of_epoch** (int): Current epoch count.
- **pre_value** (float): Last observed value of the monitored metric.
- **time_step** (int): Epochs passed since the last observed improvement in the monitored metric.

Returns:

- tuple:
  - **int**: 1 if early stopping is triggered, 0 otherwise.
  - **float**: Current monitored metric value.
  - **int**: Updated time step.
  - **int**: Cooldown flag (0 indicates no active cooldown).

Usage Example

To use EarlyStopping:



```python
early_stopping = EarlyStopping(monitor='val_loss', patience=10, min_delta=0.001, verbose=1

# During training, call `check` method each epoch to determine if training should stop
stop_flag, current_value, updated_time_step, cooldown_flag = early_stopping.check(
    metrics_values={'val_loss': [0.5, 0.4, 0.35]},
    number_of_epoch=15,
    pre_value=0.4,
    time_step=2
)
```

This class helps to automate the early stopping criteria, which is particularly useful for preventing overfitting and saving computational resources by halting training when improvements stagnate.