



Semantic Memory



Episodic Memory

of
iteration
step T



Task Initialization

Initial Prompt

🔍 As an AutoML Agent, you will be given a folder containing data and description files. Please generate Python code using autogluon. multimodal to train a predictor and make predictions on test data...

Selected Autogluon APIs

Fast and Accurate ML in 3 Lines of Code



Get Started

Tool Specific Tutorial

Tutorials

AutoMM for Semantic Segmentation - Quick Start

Semantic Segmentation is a computer vision task where the objective is to create a detailed pixel-wise segmentation map of an image, assigning each pixel to a specific class or object. This technology is crucial in various applications, such as in autonomous vehicles to identify vehicles, pedestrians, traffic signs, pavement, and other road features.

The Segment Anything Model (SAM) is a foundational model pretrained on a vast dataset with 1 billion masks and 11 million images. While SAM performs exceptionally well on generic scenes, it encounters challenges when applied to specialized domains like remote sensing, medical imagery, agriculture, and manufacturing. Fortunately, AutoMM comes to the rescue by facilitating the fine-tuning of SAM on domain-specific data.

In this easy-to-follow tutorial, we will guide you through the process of using AutoMM to fine-tune SAM. With just a single call to the `fit()` API, you can effortlessly train the model.

Prepare Data

For demonstration purposes, we use the *Leaf Disease Segmentation* from Kaggle. This dataset is a



Summarization



Condensation

Summarized Tutorials

Summary: This tutorial demonstrates implementing semantic segmentation using AutoGluon's MultiModalPredictor with SAM (Segment Anything Model). It covers essential techniques for data preparation with path handling, zero-shot inference, and model fine-tuning using LoRA. The tutorial helps with tasks like setting up semantic segmentation pipelines, performing zero-shot evaluation, and fine-tuning SAM models for domain-specific applications. Key features include configuring the predictor for foreground-background segmentation, handling data in DataFrame format, model persistence, and performance evaluation using IoU metrics. It's particularly useful for implementing efficient semantic segmentation with pre-trained SAM models while emphasizing security considerations when loading saved models.

Condensed Tutorials

```
# AutoMM for Semantic Segmentation - Quick Start
## Key Implementation Details
### Setup and Data Preparation
'''python
/pip install autogluon.multimodal

# Load and prepare dataset
import pandas as pd
import os

# Expand relative paths to absolute paths
def path_expander(path, base_folder):
    path_l = path.split('/')
    return ''.join([os.path.abspath(os.path.join(base_folder, path)) for path
in path_l])
```

Error Summary

ERROR SUMMARY:
KeyError occurs because 'optimization.batch_size' is not a valid configuration parameter in AutoGluon Multimodal's semantic segmentation configuration schema. The error indicates an incorrect hyperparameter structure in the configuration.



Retrieval

of
iteration
step T+1

Query

Retrieve

Indexing (Offline)



Task Initialization

Initial Prompt

<Task Prompt>
<Data Prompt>
<Tool Prompt>

Bash Script

```
#!/bin/bash

# Execute the Python script
CUDA_VISIBLE_DEVICES=0 python3 /xxx/generated_code.py
```

Python Code

```
'''
AutoGluon Multimodal script for camouflaged object semantic segmentation.
This script trains a model to identify and segment camouflaged objects in images.
Required packages:
pip install autogluon.multimodal
python3 -m pip install "mncv==2.1.0"
python3 -m pip install "mmdet==3.2.0"
python3 -m pip install "mengine==0.10.6"
'''
import os
import pandas as pd
import cv2
from datetime import datetime
from autogluon.multimodal import MultiModalPredictor
import numpy as np

if __name__ == "__main__":
    # Set single GPU
    os.environ["CUDA_VISIBLE_DEVICES"] = "0"

    # Define paths
```

Log & Errors

```
File ~/opt/conda/lib/python3.12/site-packages/autogluon/multimodal/learners/ba
config = get_config()
File ~/opt/conda/lib/python3.12/site-packages/autogluon/multimodal/utlis/confi
config = apply_omegaconf_overrides(config, overrides=overrides, check_key_ex
File ~/opt/conda/lib/python3.12/site-packages/autogluon/multimodal/utlis/confi
raise KeyError
KeyError: "optimization.batch_size" is not found in the config. You may need to
```

Condensed Tutorials

```
# AutoMM for Semantic Segmentation - Quick Start
## Key Implementation Details
### Setup and Data Preparation
'''python
/pip install autogluon.multimodal

# Load and prepare dataset
import pandas as pd
import os

# Expand relative paths to absolute paths
def path_expander(path, base_folder):
    path_l = path.split('/')
    return ''.join([os.path.abspath(os.path.join(base_folder, path)) for path
in path_l])
```



Error Analyzer

Error Summary

ERROR SUMMARY:
KeyError occurs because 'optimization.batch_size' is not a valid configuration parameter in AutoGluon Multimodal's semantic segmentation configuration schema. The error indicates an incorrect hyperparameter structure in the configuration.

Suggested Fix

SUGGESTED FIX:
Replace 'optimization.batch_size' with 'env.batch_size' in the hyperparameters dictionary, as batch size configuration belongs to the environment settings in AutoGluon Multimodal's configuration structure for semantic segmentation tasks.



Iterative Code Generation and Execution



Semantic Memory



Episodic Memory