AntiNex Core Worker Documentation

Release 1.0.0

Jay Johnson

Dec 07, 2018

Contents

CHAPTER 1

AntiNex Stack Status

AntiNex Core Worker is part of the AntiNex stack:

Component	Build	Docs Link	Docs Build
REST API		Docs	
Core Worker		Docs	
Network Pipeline		Docs	
AI Utils		Docs	
Client		Docs	

CHAPTER 2

Table of Contents

These are the docs for the AntiNex Core Worker repository.

2.1 Source Code

2.1.1 AntiNex Core Worker - API Reference

2.1.2 Splunk Environment Variables

This repository uses the Spylunking logger that supports publishing logs to Splunk over the authenticated HEC REST API. You can set these environment variables to publish to Splunk:

```
export SPLUNK_ADDRESS="<splunk address host:port>"
export SPLUNK_API_ADDRESS="<splunk api address host:port>"
export SPLUNK_USER="<splunk username for login>"
export SPLUNK_PASSWORD="<splunk password for login>"
export SPLUNK_TOKEN="<Optional - username and password will login or you can use a␣
↪pre-existing splunk token>"
export SPLUNK_INDEX="<splunk index>"
export SPLUNK_QUEUE_SIZE="<num msgs allowed in queue - 0=infinite>"
export SPLUNK_RETRY_COUNT="<attempts per log to retry publishing>"
export SPLUNK_RETRY_BACKOFF="<cooldown in seconds per failed POST>"
export SPLUNK_SLEEP_INTERVAL="<sleep in seconds per batch>"
export SPLUNK_SOURCE="<splunk source>"
export SPLUNK_SOURCETYPE="<splunk sourcetype>"
export SPLUNK_TIMEOUT="<timeout in seconds>"
export SPLUNK_DEBUG="<1 enable debug|0 off - very verbose logging in the Splunk␣
↪Publishers>"
```

2.1.3 Celery Worker

Here is the Celery Worker's source code.

Process Consumed Messages From the Queues

The processor class processes any messages the worker consumes from the queue.

Send Results to the Broker

This method is responsible for publishing what the core's results were from the processed job.

Note: The results must be sent back as a JSON dictionary for the REST API's Celery Workers to handle.

2.1.4 Scripts

Standalone Processing Examples

Using Scaler Train and Test Helper

Train a DNN using the Scaler-Normalized AntiNex Django Dataset. This builds the train and test datasets using the antinex_utils.build_scaler_train_and_test_datasets.py method from the internal modules.

Using Manual Scaler Objects

Train a DNN using the Scaler-Normalized AntiNex Django Dataset. This builds the train and test datasets manually to verify the process before editing the antinex_utils.build_scaler_dataset_from_records.py method.

antinex_core.scripts.standalone_scaler_django.**build_model**(*num_features*, *loss*, *optimizer*, *metrics*)

Build the Keras Deep Neural Network Model

Parameters

- **num_features** – number of features
- **loss** – loss function to apply
- **optimizer** – optimizer to use
- **metrics** – list of metrics

antinex_core.scripts.standalone_scaler_django.**build_a_scaler_dataset_and_train_a_dnn**(*name='set'*)

Build a scaler-normalized dataset and then build a deep neural network for training and predictions

Parameters **name** – name for the dnn

Convert Bottom Rows from a CSV File into JSON

When testing live DNN predictions you can use this utility script to print a few JSON-ready dictionaries out to `stdout`.

Usage:

```
convert_bottom_rows_to_json.py -f <CSV File> -b <Optional - number of rows from the
↪bottom>
```

S3 Testing

Run this script to verify S3 is working.

Set Environment Variables

Set these as needed for your S3 deployment

```
export S3_ACCESS_KEY=<access key>
export S3_SECRET_KEY=<secret key>
export S3_REGION_NAME=<region name: us-east-1>
export S3_ADDRESS=<S3 endpoint address host:port like: minio-service:9000>
export S3_UPLOAD_FILE=<path to file to upload>
export S3_BUCKET=<bucket name - s3-verification-tests default>
export S3_BUCKET_KEY=<bucket key name - s3-worked-on-%Y-%m-%d-%H-%M-%S default>
export S3_SECURE=<use ssl '1', disable with '0' which is the default>
```

Run S3 Verification Test

Run the included S3 verification script:

```
run-s3-test.py
```

S3 verification tool

antinex_core.scripts.run_s3_test.**run_s3_test**()
Run the S3 verification test

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index