DeepAR Time Series Forecasting

Content Prepared By: Chandra Lingam, Cloud Wave LLC

DeepAR Forecasting

DeepAR is a time series forecasting algorithm

Supervised algorithm based on Recurrent Neural Networks

Typical Use – Forecast:

- Product demand
- Passenger traffic
- Weather trend
- And more

Time component

Almost all data has a time component to it:

ATM transaction, Social Media Post, Doctor appointment, Departure/Arrival Time

Are these considered time series?

Time series

“a series of values of a quantity obtained at successive times, often with equal intervals between them.”

Time Series Forecasting with non-time series algorithms

Do not natively handle Date Time Features

Do not learn from time dependent nature of data – rather it is completely feature driven

DeepAR

Classical forecasting methods (ARIMA or ETS) fit a single model to each time series

DeepAR can fit many similar time series in a single model

DeepAR outperforms ARIMA/ETS on datasets that contain hundreds of related time series

DeepAR Time Series Forecasting

Train model with one or more time series

Use model to extrapolate time series into future

Use model to generate forecasts for a new time series that is similar to the ones it is trained on

prediction_length hyperparameter determines how many points in future needs to be forecast

DeepAR Stationary vs Non-Stationary

Classical Time series forecasting algorithms expect time series to be stationary (mean, standard deviation are constant over time)

- You would need to transform the data to make it stationary

DeepAR does not expect data to be stationary. You can train with data as-is.

- Verify by experiments if stationarity improves model

Training/Test Important Differences

With other algorithms:

- Data is randomly divided into training and test sets
- Training file contains only training data
- Test file contains only test data
- Inference: Model predicts target for new data

DeepAR Training/Test

With DeepAR:

- Time series is split based on time – time order needs to be honored
- Training – Entire dataset except for last prediction_length points
- Test – Entire dataset

DeepAR – Data Format

Input:

JSON Lines (.json, .json.gz)

Parquet (.parquet)

Inference:

JSON

DeepAR Training Input

Field	Description
start	Start Timestamp for the time series Format: YYYY-MM-DD HH:MM:SS
target	Array of floating point or integer values in a time series Supports missing values: “NaN” in JSON Lines, nan in Parquet
dynamic_feat	Optional input features. Floating point or integer values. Array of Arrays. Each inner array represents values of a feature and must be of same length as target Missing values are not supported in the features
cat	Optional category. Category is used for identifying/encoding a time series. Each categorical feature is represented as 0 based integer. If you are using categories, then training set must have timeseries for all categories. DeepAR can forecast only for categories that were trained with.

DeepAR Inference Format

Field	Description
instances	Corresponds to time series that should be forecast
start	Start Timestamp for the time series Format: YYYY-MM-DD HH:MM:SS
target	Array of values. Time series for which to forecast
dynamic_feat	Field should be included only if model was trained with features. You must provide dynamic features for each value in target and for future timepoints Length(dynamic_feat) = length (target) + prediction_length
cat	Field should be included only if model was trained with categories.

Demo – Time Series

Time series with Pandas

Detecting missing time steps

Handling missing values

Demo - Overview

- Kaggle Bike Rental as a time series forecasting problem
- How to handle missing values in time series
    - Training Data consists of two years worth of hourly rental
    - Gaps in time series - Training data consists of first 19 days of each month
- How to handle missing features

Demo

Demo 1 - Train only with target time series

Demo 2 - Add Categories

Demo 3 – Add Dynamic Features

Note: Demos focus on how to get things done with DeepAR. Emphasis is not on best Kaggle scores

AWS Provided Examples: Synthetic Data, Electricity Forecast

Demo 3 – Dynamic Features

Requires More Powerful Training instance

Trained with ml.c5.4xlarge instance (16 CPU, 32 GB RAM)

Training with ml.m4.xlarge (free-tier) or ml.m5.xlarge – Out of memory exception (4 CPU, 16 GB RAM)

For Endpoint, you can still use ml.m4.xlarge instance

Total Cost: USD 0.50 (for training job) + USD 0.20 (endpoint – free tier eligible)

DeepAR Hyperparameters

[DeepAR Hyperparameters]

[Parameters with greatest impact]

Useful Resources

Beginner’s Guide to create a Time Series Forecast by Aarshay Jain

Handling Missing Values in Time Series For Beginners by jingjuewang

Pandas: Working with missing data

DeepAR: How it works

DeepAR: Modeling non-stationary data Q&A