HIS PROJECT - TSA

Task 08

Author
Alberto Mejia

Contents

1 Chapter 15: Strategies for Global Deep Learning Forecasting Models **3**
1.1 Creating global deep learning forecasting models 3
1.2 Using time-varying information 3
1.3 Using static/meta information 3
1.4 Using the scale of the time series 4
1.5 Balancing the sampling procedure 4
1.6 Summary 5

1 Chapter 15: Strategies for Global Deep Learning Forecasting Models

1.1 Creating global deep learning forecasting models

Chapter 10 emphasizes the importance of global forecasting models in the context of deep learning. Key points include:

- Global models offer advantages like increased data size, cross-learning, multi-task learning, regularization, and reduced engineering complexity.
- Deep learning models require substantial data, making global models a compelling choice.
- Transitioning from individual to global models is facilitated by adapting the data loader to sample from multiple time series.
- The PyTorch Forecasting library simplifies time series forecasting with deep learning, offering a high-level API and a TimeSeriesDataset class for data preparation.

In summary, Chapter 10 highlights the benefits of global models in deep learning forecasting and introduces tools like PyTorch Forecasting to streamline model development. In the first implementation, there is only one decoder in each timestep and target sequence. The model is trained exclusively on the historical data, and its task is to predict future values based solely on this historical information. This approach simplifies the forecasting process, making predictions based on the past without considering complex multi-sequence interactions or external factors.

1.2 Using time-varying information

In this context, we face a challenge when setting up the model due to the nature of historical data. We need to consider that time-varying known features and the history of the target variable cannot be used together for single-step-ahead forecasting. This is because the time-varying features have future values known in advance, making them suitable for prediction at a given timestep. In contrast, the history of the target variable lacks this advantage, as it represents the quantity we aim to forecast, leading to a causal constraint. The current tensor arrangement aligns these two types of data on the same timesteps, which needs to be addressed to avoid inappropriate forecasting.

1.3 Using static/meta information

Incorporating household-specific features, like the Acorn group and dynamic pricing, into the model allows it to capture unique patterns associated with different households. Categorical features, which are non-numeric, can pose challenges in machine learning models. However, in deep learning models, there

is a distinct approach for handling these categorical features called embedding vectors. This technique involves converting categorical variables into continuous vector representations, making them suitable for neural networks. This allows the model to learn complex relationships and patterns within categorical data, enhancing its forecasting capabilities.

- **One-hot encoding:** Transforms categorical data into a high-dimensional binary format, with each category becoming a dimension. This method creates a sparse representation, adding dimensions equal to the category count. However, it treats all categories as equally distant, disregarding potential similarities. For instance, it fails to capture that Saturday and Sunday are closer as weekend days.

- **Embedding vectors:** offer a dense representation of categorical features through an embedding layer, which maps each category to a numerical vector. These vectors have lower dimensions than the categorical feature's cardinality. During training, the model determines the optimal vector for each category. This approach is common in natural language processing, where it's used to embed thousands of words into dimensions as small as 200 or 300.

1.4 Using the scale of the time series

To scale time series data, GroupNormalizer in TimeSeriesDataset is used to make the target zero mean and unit variance. This prevents the model from adjusting its parameters based on individual household consumption scales. However, it results in information loss because unique patterns in larger and smaller consumption households are combined. To reintroduce this scale information without including unscaled targets, mean and standard deviation data for each household, collected during the initial scaling process, can be added as static-real features to the model. This provides the model access to scale information while learning time series patterns.

1.5 Balancing the sampling procedure

The analogy of the bowl and chits of papers is used to explain the concept of balancing the sampling procedure in the context of selecting batches from time series data. Imagine the bowl filled with balls, where each ball represents a time series (household), and the chits of paper represent different windows of samples that can be drawn from each time series.

In the default batch sampling procedure, all the chits are dumped into the bowl, and a batch is randomly selected by picking a certain number of chits. This can result in a biased batch composition, especially if there is an imbalance in the characteristics of the time series, such as different lengths or consumption levels.

To address this bias, a threshold-based sampling approach is introduced. Instead of uniformly selecting chits, a random number between 0 and 1 (p) is

generated for each chit, and the chit is selected only if p is less than a certain threshold (e.g., p ¡ 0.5). By adjusting this threshold, it becomes possible to control the acceptance rate of chits, making it harder or easier for a chit to be included in the batch.

The goal is to find the right thresholds for each chit so that the resulting batch has a more balanced representation of different characteristics (represented by colors in the analogy). To achieve this, the thresholds are determined based on the inverse of the count of chits in each category or bin. Bins with fewer samples get higher weights, making it more likely for their chits to be selected, while bins with more samples have lower weights, reducing their chances of inclusion. This strategy helps create batches with a more even distribution of different characteristics.

1.6 Summary

In recent chapters, we've laid the groundwork for deep learning models and explored the concept of global models in deep learning. We've utilized PyTorch Forecasting and the versatile TimeSeriesDataset for model development.

Starting with a basic LSTM in the global context, we expanded our models by incorporating time-varying, static, and individual time series scale information, enhancing their performance. We also delved into an alternating sampling approach for mini-batches to ensure balanced representation in each batch.

While this chapter provides valuable techniques for improving forecasting models, it serves as a foundation for further model development. In the upcoming chapter, we'll explore specialized deep learning architectures tailored for time series forecasting.