Timo Meiendresch

Recurrent Neural Networks for Time Series Forecasting

Capstone Project
Machine Learning Engineer Nanodegree (Udacity)

Cologne 2019

Contents

List of Figures

List of Tables

Abstract

The year is 2019 A.D. Quantitative research is entirely occupied by Machine Learning (ML). Well, not entirely... One small group of indomitable forecasters still holds out against the invaders. And life is not easy for the ML evangelists who garrison the fortified camps of quantitative research.

1 Introduction

It is widely perceived, that time series forecasting is an essential tool in business, economics, and many other disciplines. The ability to accurately infer future variables given historic data is of huge value to decision makers.

In recent years, **Machine Learning (ML)** methods had huge success in many areas of quantitative research. Prominent examples include classification tasks, image processing, or text analysis. ML took many fields by storm and it seems that many quantitative discipline can benefit from giving in to the AI wave.

And yet, in the area of time series forceasting, machine learning methods are rarely considered. In this area they have a history of inferior performance compared to simpler, traditional methods. Particular examples of these traditional methods include the **Autoregressive Integrated Moving-Average (ARIMA)** and **Exponential Smoothing (ES)** methods.

Until recently, a widely shared perception among researchers was that complex methods for time series forecasting in general, such as neural networks, were not performing better than traditional ones (e.g. Hyndman, 2019). Among others, Makridakis et al. (2018) noted that there is only very limited scientific evidence which suggests that artificial neural networks may be a useful tool for time series forecasting.

Paradigm Shift

But, advances in recent years start to challenge this notion. For example, the winner of the most recent version of the M4 time series competition (Makridakis et al., 2019) was an elaborate hybrid of a **recurrent neural network (RNN)** architecture with ES methods (Smyl, 2019). Measured on a diverse dataset of 100,000 univariate time series from six domains and frequencies, this method outperformed traditional ones, pure statistical combinations as well as ML-based combination methods. The results of the competition showed the potential of recurrent networks for time series forecasting, leading to a surge of algorithms in this area. It also showed that RNN could successfully be trained on highly diverse data. Commonly, RNN methods for time series foreasting are applied to a large set of homogeneous data.

Local vs. global methods

Moreover, the M4 competiton highlights an ongoing paradigm shift in the forecasting community. Traditional methods like ARIMA and ES are applied to individual series. These **local** methods estimate a number of parameters within a model space that is restricted by the respective structure of the model. Because these approaches rely on an explicit model structure they are referred to as **model-based** (Wang et al., 2019). Each series is modeled and trained independent of other series in the data set. Hence, the individual model is independent of the total number of series in the dataset as well as its characteristics in terms of similarity.

Also, with the emergence and availability of large sets of data a new forecasting problem emerges. It is becoming increasingly necessary to implement large-scale forecasting systems that simultaneously predict many related series. Examples include predictions of product sales in online retailers, household energy consumption, or web traffic. In contrast to local models, **global** methods enable the use of **cross-series learning**, i.e. the training process uses all available time series and yields a representation of the entire data set. In theory, RNN-based methods are therefore able to learn across individual series and extract general patterns of the data as well as learn dependency relationships between the individual series which may improve forecasting accuracy.

In the aftermath of the M4 competition, various methods have been published that are based on the idea of cross-series learning, often combining local with global methods. If and under which conditions these models actually perform better than local ones is a relatively new question into which very little empirical research has so far been done.

In theory, RNN should be able to work as a universal function approximator if it is only powerful enough. In practice however, the learnability of RNNs is limited. A typical problem is that they do not converge during training. They are considered to be difficult to train. This is an important factor for practitioner's as training a powerful network on such a huge amount of data is computationally expensive.

Hence, it seems that there are two main properties of the underlying data that should be considered in the following analysis:

> Property 1 - Total number of individual time series in the underlying data set. How many series are in the dataset?
>
> Property 2 - Similarity of the individual time series in the data set. How diverse are the series?

This leads to two hypothetically important caveats regarding the dataset which influence the performance. The total number of time series in the dataset as well as how diverse the series are with respect to each other.

Often these methods lack rigorous comparison with previous methods and known data sets. This capstone project investigates the performance of three RNN-based methods developed in the last two years, namely:

- **Deep Autoregressive Recurrent Neural Networks (DeepAR)** - Wang et al. (2019)
- **Deep State Space Models (DeepState)** - Rangapuram et al. (2018)
- **Deep Neural Network Factor Models (DeepFactor)** - Wange et al. (2019)

To evaluate these models I will apply them to the Hourly, Daily, Weekly and Monthly data of the M4 competition using the same metrics that were used in the competiton. Hence, a variety of benchmarks is available for comparison.

Note that these methods are designed to be applied to large sets of related data. In contrast, the M4 data set is a diverse dataset from many domains and frequencies.

To the best of my knowledge, no comparison of these algorithms to the M4 benchmarks has been published yet. The primary contribution of this paper is to evaluate these algorithms on the well studied M4 data and compare the results to known benchmarks.

The rest of the paper is organized as follows. I first discuss ... in Section ....

2 Definition

Project Overview

This project compares three RNN-based algorithms for time series forecasting to the well studied M4 data and its benchmarks, including modern and traditional methods. The data include the original M4 data for hourly, daily, weekly, and monthly frequencies.

Problem Statement

Main goal of this paper is to evaluate the performance of the three algorithms DeepAR, DeepState, and DeepFactor with respect to traditional and state-of-the art methods. The main question is to give a quick assessment of their competitiveness starting from their default hyperparameters. The accuracy is assessed using the same metrics that were used during the competition.

Metrics

The performance of the methods is evaluated using the same two measures that were used in the M4 competition to evaluate the forecast accuracy. Forecast measures are still a highly active area of research and there is no consensus on which one is considered to be the best one. Hence, two widely used **scale-independent errors** have been used in the competition. Scale-independent means they are independent of the unit of the time series. Because the competition evaluates accuracy across highly diverse time series, a scale-independent measure is required. The two measures are:

- **Symmetric mean absolute percentage error (sMAPE)**
- **Mean absolute scaled error (MASE)**

The **sMAPE** belongs to the class of percentage errors. While this makes the measure unit-free it has the disadvantage of being infinite/undefined if the target variable is zero for any t in the period of interest, and is highly skewed towards extreme values if the target variable is close to zero. The measure is defined as

$$sMAPE = \frac{1}{h}\sum_{t=1}^{h}\frac{2\,|Y_t - \hat{Y}_t|}{|Y_t| + |\hat{Y}_t|}$$

where Y_t is the post-sample value, $\hat{Y}_t$ the estimated forecast, and h the forecasting horizon.

Alternatively, Hyndman and Koehler (2006) propose the scaled error measure **MASE** when comparing forecast accuracy across series with different units. In this case the absolute error is scaled using the **Mean Absolute Error (MAE)** from a simple forecast method (either naive or seasonal naive method). The method is defined for seasonal in the following way:

$$MASE = \frac{1}{h}\frac{\sum_{t=1}^{h}|Y_t - \hat{Y}_t|}{\frac{1}{n-m}\sum_{t=m+1}^{n}|Y_t - Y_{t-m}|}$$

where m indicates the seasonality of the series. In case of non-seasonal data $m = 1$, $Y_t - Y_{t-m}$ simplifies to $Y_t - Y_{t-1}$ being the naive forecast error. The naive forecast predicts the next value to be equal to the last value $\hat{Y}_{t+1} = Y_t$. This yields the non-seasonal version of the MASE as

$$MASE = \frac{1}{h}\frac{\sum_{t=1}^{h}|Y_t - \hat{Y}_t|}{\frac{1}{n-m}\sum_{t=m+1}^{n}|Y_t - Y_{t-m}|}$$

In addition to evaluating point forecasts, the performances of prediction intervals has

also been part of the competition. Specifically, the **Mean Scaled Interval Score (MSIS)** has been used for this. Moreover, other widely used measures for evaluating forecast accuracy include the normalized sum of quantile losses for quantiles $p \in 0.5, 0.9$. As these were the preferred measures in some of the mentioned papers I have decided to include them for reference together with the MSIS. However, there is high correlation between the two forecast measures and MSIS as well as there are some methods that do not give quantile losses. Therefore, I will not discuss these results explicitly.

3 Analysis

Data Exploration

The dataset consists of 100,000 univariate time series from six domains, divided in six different frequencies. They were chosen from a database of more than 900,000 time

M4 Data - Overview

Frequency	Micro	Industry	Macro	Finance	Demographic	Other	Total
Yearly	6,538	3,716	3,903	6,519	1,088	1,236	23,000
Quarterly	6,020	4,637	5,315	5,305	1,858	865	24,000
Monthly	10,975	10,017	10,016	10,987	5,728	277	48,000
Weekly	112	6	41	164	24	12	359
Daily	1,476	422	127	1,559	10	633	4227
Hourly						414	414
Total	25,121	18,798	19,402	24,534	8,708	3,437	100,000

Table 1: M4 data by domains and frequencies.

series. One time series consists of a date and a respective observation for that date where the frequency determines the space between two observations. In this dataset, all frequencies are equidistant which means that the time between two observations is constant.

Common characteristics include seasonal patterns, cyclic behavior, as well as trends that the model has to capture. Given the behavior of the series, the method has to extrapolate the series into the future. The number of future observations that have to be forecasted were given by the instructions of the competitions: A key challenge for the prediction method is the high diversity of the data. Some data exhibit cyclical behavior, whereas other series have up- or downward trends, or varying seasonal patterns. Capturing the behavior accurately for all series is the goal of time series forecasting.

M4 Data - Forecast Horizons

Frequency	Forecast horizon
Yearly	6
Quarterly	8 (2 years)
Monthly	18 (1.5 years)
Weekly	13 (3 months)
Daily	14 (2 weeks)
Hourly	48 (2 days)

Table 2: Forecast horizons by frequency of the data.

Exploratory Visualization

The dataset is highly diverse with respect to domain and frequency. For each series there exist a train data as well as extended test data. The test data consists of the train data and the additional, previously unknown observations that have to be forecasted. A randomly chosen example series from the hourly M4 dataset is depicted below:

M4 Hourly - Series #338

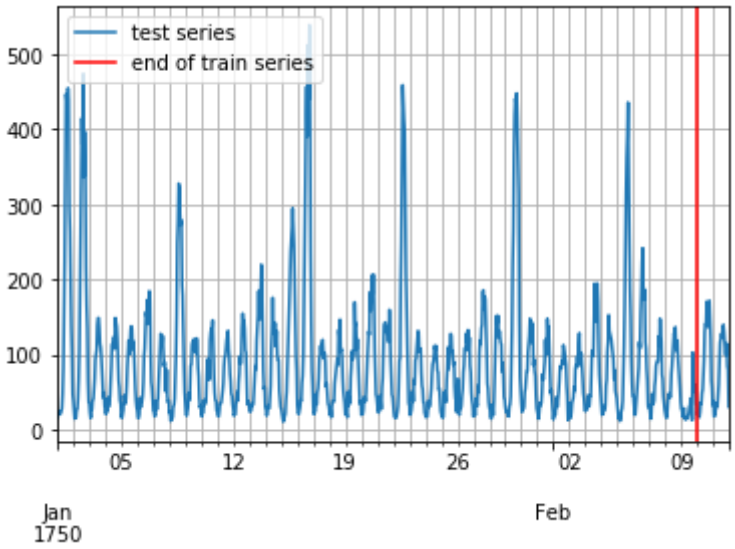

Figure 1: Example series over time. The vertical red line indicates the break between train and test dataset.

Algorithms and Techniques

In this paper I will compare three RNN-based algorithms to the results of the M4 competition.

The first of the three algorithms is the **Deep Autoregressive Recurrent Neural Network (DeepAR)** method, introduced by Salinas et al. (2017). It is the only

built-in time series prediction algorithm in the AWS SageMaker which enables machine learning in the cloud. DeepAR uses autoregressive features that are used as the inputs to a recurrent neural network. The RNN uses LSTM cells as default but can also be changed to GRU cells. Previous observations are taken in as inputs which makes this algorithm autoregressive. Which autoregressive features are included depends on the frequency of the data.

Rangapuram et al. (2018) introduced **Deep State Space Models (DeepState)**. DeepState combines state space models (SSM) with a recurrent neural network architecture that calculates the optimal parameters for the SSM. The SSM is applied locally to the individual time series. The parameters are determined by a RNN which is trained jointly on all available series.

Furthermore, I included the **Deep Factor** model by Wang et al. (2019). This hybrid algorithm uses local methods that are specific to each time series in the dataset as well as a global representation trained on the entire data. For now, the only local method that is available is a smaller RNN. Hence, the model combines a local RNN with a global RNN (DeepFactor-RNN or DF-RNN in the following). Main differences between the algorithms:

- **DeepAR** - Pure RNN that uses autoregressive features as inputs.
- **DeepState** - Uses RNN to determine the best parameters of the individual SSM
- **DF-RNN** - Hybrid of a local RNN (individual time series) that is combined with a global RNN (all time series)

The training process for all three algorithms uses a fixed window approach. During training, batches of time series windows in the training data are fed to the algorithm. Each time series window is a sliced version of the entire series starting at different points of the series and has length of the training range plus prediction range. Here, the prediction range corresponds to the forecasting horizon of our model.

Benchmark

In addition to providing a rich dataset across frequencies and domains, a key contribution of the M4 competition is the provision of a rich set of benchmark methods. Not only the metrics of the competition winners but also results for commonly used methods, in particular ARIMA and ES, are provided in Hyndman et al. (2019). In the discussion of the results I will focus on the relative performance of the RNN-based methods with respect to:

1. seasonally adjusted naive predictions (Naive2)

2. traditional methods (ARIMA and ES)

3. overall winning method of Smyl (2019), runner-up by Montero-Manso et al. (2019) as well as the winning method of the respective frequency

4 Methodology

Data Preprocesing

The data has to be prepared in a certain way to be valid inputs to the algorithms. Namely, it is required to be in json lines like format which in python is essentially a dictionary of dictionaries. Inputs must contain at least the following keys with respective content:

- 'start' - Timestamp with the format YYYY-MM-DD HH:MM:SS and the respective frequency of the data
- 'target' - An array of floats or integers that represents the target series of interest.

An example input file takes the form

```
# get first entry in dataset.train
entry = list(dataset.train)[0]

# first entry
entry
>>> {'start': Timestamp('1750-01-01 00:00:00', freq='H'),
 'target': array([605., 586., 586., ..., 521.], dtype=float32)
```

Additional fields are optional. One important field

Implementation

Refinement

5 Results

Model Evaluation and Validation

Justification

6 References

[1] Hyndman, Rob J. "A brief history of forecasting competitions." *International Journal of Forecasting (2019).*

[2] Hyndman, Rob J., and Anne B. Koehler. "Another look at measures of forecast accuracy." *International journal of forecasting 22.4 (2006): 679-688.*

[3] Rangapuram, Syama Sundar, et al. "Deep state space models for time series forecasting." *Advances in Neural Information Processing Systems*. 2018.

[4] Salinas, David, Valentin Flunkert, and Jan Gasthaus. "DeepAR: Probabilistic forecasting with autoregressive recurrent networks." *arXiv preprint arXiv:1704.04110 (2017).*

[5] Smyl, Slawek. "A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting." *International Journal of Forecasting (2019).*

[6] Spiliotis, Evangelos, et al. "Are forecasting competitions data representative of the reality?." *International Journal of Forecasting (2019).*

[7] Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. "Statistical and Machine Learning forecasting methods: Concerns and ways forward." *PloS one 13.3 (2018): e0194889.*

[8] Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. "The M4 competition: 100,000 time series and 61 forecasting methods." *International Journal of Forecasting (2019).*

[9] Wang, Yuyang, et al. "Deep Factors for Forecasting." *arXiv preprint arXiv:1905.12417 (2019).*