PROGRESSIVE GROWING OF NEURAL ODES

Hammad A. Ayyubi *
University of California, San Diego
hayyubi@ucsd.edu

Yi Yao & Ajay Divakaran
SRI International
{yi.yao,ajay.divakaran}@sri.com

ABSTRACT

Neural Ordinary Differential Equations (NODEs) have proven to be a powerful modeling tool for approximating (interpolation) and forecasting (extrapolation) irregularly sampled time series data. However, their performance degrades substantially when applied to real-world data, especially long-term data with complex behaviors (e.g., long-term trend across years, mid-term seasonality across months, and short-term local variation across days). To address the modeling of such complex data with different behaviors at different frequencies (time spans), we propose a novel progressive learning paradigm of NODEs for long-term time series forecasting. Specifically, following the principle of curriculum learning, we gradually increase the complexity of data and network capacity as training progresses. Our experiments with both synthetic data and real traffic data (PeMS Bay Area traffic data) show that our training methodology consistently improves the performance of vanilla NODEs by over 64%.

1 INTRODUCTION

Time series analysis is critical in a number of domains such as stock prices analysis, weather analysis, business planning, resource allocation, etc. One major aspect of such time series analysis is dealing with irregularly sampled data. Prior approaches tackle this issue by mapping such data onto equally spaced intervals (Lipton et al., 2016). However, this approximation introduces error, especially at the local maxima and minima of seasonal fluctuations.

Several approaches have been proposed to improve approximation accuracy. Mei & Eisner (2016) uses exponential decay to model state between observations. Neural Ordinary Differential Equations (NODEs) Chen et al. (2018) model continuous states between observations using a continuous depth black box ODE solver parameterized by a neural network. NODEs have proven to be promising for forecasting problems with irregular samples (Rubanova et al., 2019). However, they are brittle when tasked with forecasting functions containing long-term trends (yearly) and short-term seasonalities (monthly or daily).

To address the above mentioned issue, we propose novel networks based on Progressive Neural Ordinary Equations (PODEs). Specifically, we follow a curriculum learning approach in which we gradually increase the data complexity as well as network complexity as training progresses. The key idea is that the network learns low frequency and easier to learn trends first and then the high frequency and more complex seasonalities. Such a breakdown of task enables the network to gradually learn these complex curves, which is, otherwise, too difficult to learn.

We summarize the contribution of the paper as follows:

- We propose novel Progressive Neural ODEs (PODEs) for the analysis of irregularly sampled complex time series containing trends and seasonalities.
- We demonstrate empirical evidence of the superiority of our approach as compared to vanilla NODEs on both synthetic and real-world data.

2 RELATED WORK

Time Series Modeling   Compared with the extensive body of work on time series forecasting of regularly sampled (i.e., equally-spaced) data (Box et al. (2015); Brockwell & Davis (2016)), fewer methods exist for irregularly sampled (i.e., unevenly-spaced) data. Analysis of such data becomes a critical challenge associated with complex real-world applications such as economics (Harvey &

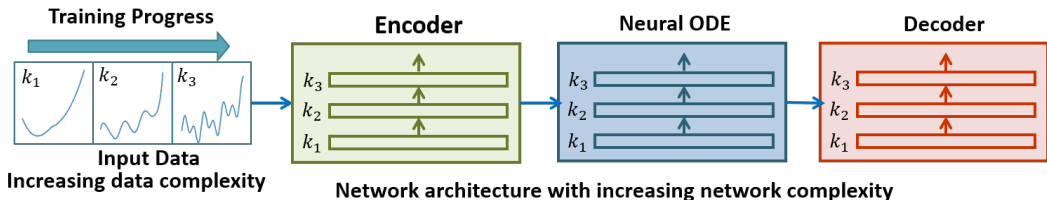*This work was done while the author was interning at SRI International.

Figure 1: Overview of our approach. Following curriculum learning approach, data complexity and network complexity is gradually increased (from k_1 to k_3) as training progresses.

Todd (1983)), healthcare (Li & Marlin (2016)), and astronomy (Scargle (1982)), to name a few. One major line of methods transform irregularly spaced samples into equally spaced ones and then apply existing methods for equally spaced data (Zhang (2003)). For instance, Gaussian Process combined with learned neural networks is recently applied for interpolating irregularly sampled data (Li & Marlin (2016); Shukla & Marlin (2018)). However, such methods suffer from a number of biases (Rehfeld et al. (2011)), which significantly degrades the overall performance, especially for highly irregular observations. Classical exponential smoothing methods such as Holt and Winters (Gardner Jr (1985); Holt (2004)) are applied to irregularly-sampled time series mainly for the estimation of trends and seasonals. With the significant development in deep learning, learned networks in a data-driven manner (e.g., NODEs) also find promising applications to analysis of irregularly sampled data.

Curriculum Learning The idea that humans learn in an organized manner, leveraging previous experience/knowledge to learn more complex tasks, originates from cognitive science. Elman (1993) first explored whether this idea can be used to train neural networks. He showed how curriculum learning can be used to learn simple language grammar. Bengio et al. (2009) extended this idea and showed it's effectiveness in language modeling and geometric shape recognition task. More recently, this idea of progressive learning has been explored in many contexts. Zaremba & Sutskever (2014) use this approach for evaluating short computer programs. Matiisen et al. (2017) apply curriculum learning in reinforcement learning regime, where a student learns a complex task following a teacher's direction of learning subtasks. Karras et al. (2017) show impressive performance in generating human faces using Generative Adversarial Networks (GANs) following the same learning strategy. Inspired by this line of works, we plan to apply progressive learning strategy to NODE-based time series forecasting in order to improve its performance on complex real-world data with trends and seasonalities.

3 OUR APPROACH

To illustrate the advantages of PODEs, we choose to use the same network design originally proposed in Rubanova et al. (2019) as our backbone architecture. This architecture consists of an encoder, a NODE, and a decoder as shown in fig. 1. The encoder maps the input data into a fixed length embedding. The NODE network models the temporal dynamics of the data using irregularly spaced samples and make prediction of future values. Finally, the decoder transforms the prediction represented in the latent embedding space to actual output.

The fact that this architecture, despite it's theoretical advantage, failed to model complex time-series functions containing trends and seasonalities (section 4.1 and 4.2), prompted us to employ a progressive learning approach. Under this scheme, we reorganize network layers into groups and train each group of layers progressively using data with gradually increasing complexity. The key idea is that we divide the complex task of learning functions containing trends and seasonalities into much easier to learn sub-tasks.

The network architecture and training procedure is illustrated in fig. 1. We divide the training stages into k steps. At each step, we add a group of layers to the encoder, NODE and decoder. Concurrently, at each step we increase the complexity of input data. All the network layers - both the previously trained and the newly added ones - remain trainable throughout training. To alleviate instability introduced by adding new layers, we use alpha blending - gradual addition of the new layer controlled by the parameter α. We prepare the input data for the $1, ..., k-1$ steps using $k-1$ low pass filters. Note that the original data is used as the input for the kth step.
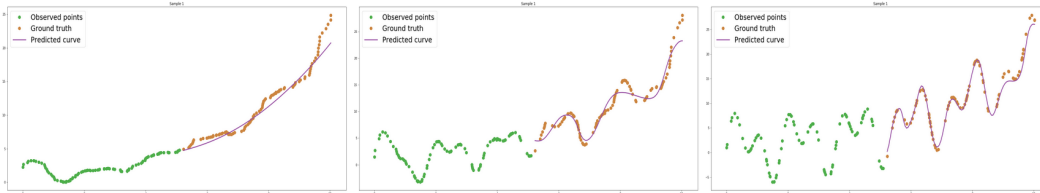
Figure 2: Forecasting performance at different training stages. Left: $k = 1$. Middle: $k = 2$. Right: $k = 3$. As training progresses, data complexity along with network complexity increases enabling the learning of complex time series.Green dots: irregularly sampled observations. Orange dots: ground truth. Purple curve: prediction.
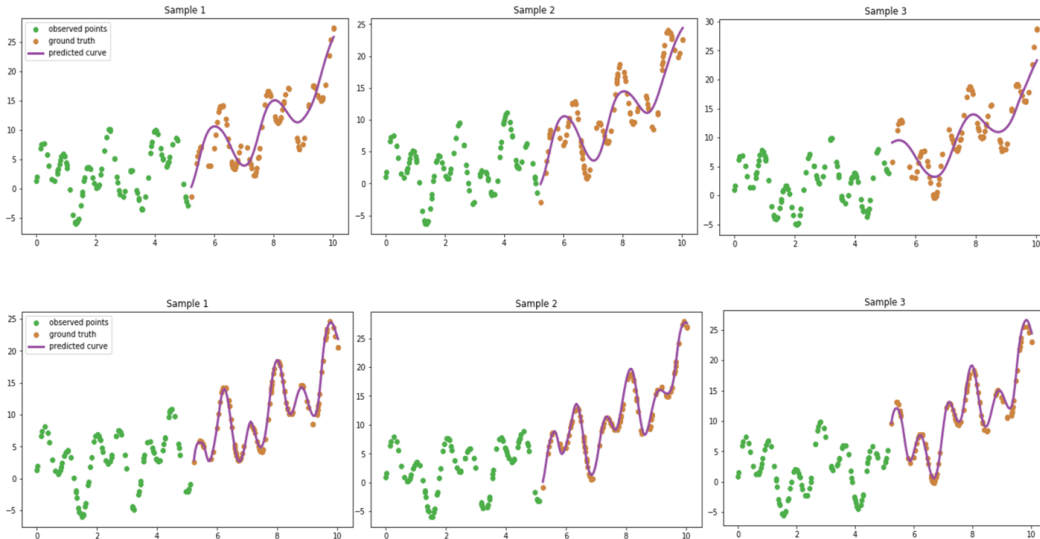

Figure 3: Forecasting performance comparison between NODE (top) and PODE (bottom) on synthetic data - $exp(cx) + sin(t_1 x) + sin(t_2 x)$. Each column shows samples with different c for trends and t_1, t_2 for seasonal fluctuations. Overall speaking, NODE can capture trend (c) and one seasonal component with lower frequency (t_1) whereas PODE can capture not only trend (c) but also both seasonalities (t_1 and t_2). Green dots: irregularly sampled observations. Orange dots: ground truth. Purple curve: prediction.

4 EXPERIMENTS

We use a Gated Recurrent Unit as encoder and a feed forward network as the decoder. We train our network in $k = 3$ steps. At each step, we add a layer. Therefor, we have three layers in total in each of our sub networks. To be consistent with NODEs, we use the same hyperparameters as described in Rubanova et al. (2019) - an initial learning rate of 1e-2 with an exponentially decaying schedule and a batch size of 50.

We compare the performance of PODE with respect to the original NODE Rubanova et al. (2019) using two datasets: a synthetic datasets for experiments with controlled parameters such as trend rate and seasonal frequencies and a real-world dataset - the PeMS-Bay traffic datasets, a commonly used dataset for time series forecasting. We also compare our model to traditional approaches: (1) Static Model: predicts the same value as encountered p time steps before, (2) Historical Average (HA): predicts weighted average of past seasons as its forecast, and (3) ARIMA: Auto-Regressive Integrated Moving Average which is an auto-regressive model popularly used for time series prediction.

4.1 SYNTHETIC DATA

We generate synthetic data of the form $exp(cx) + sin(t_1 x) + sin(t_2 x)$, where c, t_1, t_2 control the global trend and local seasonals. This is a simplified version of real-world data with trends (i.e.,

Dataset	Models				
	Static	HA	ARIMA	NODE	PODE
Synthetic	36.43	35.74	29.69	15.56	**0.81**
PEMS-BAY (+E03)	17.58	40.45	47.78	13.80	**4.87**

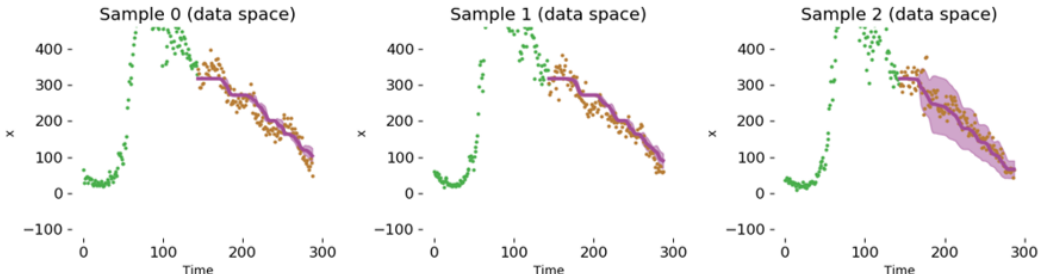Table 1: MSE on synthetic and PEMS-BAY datasets.

Figure 4: Forecasting performance PODE on PeMS-BAY. Each column shows samples from different sensors. Green dots: irregularly sampled observations. Orange dots: ground truth. Purple curve: prediction. Purple shaded area: prediction uncertainty.

the exp function) and multiple seasonalities (i.e., the two sin functions). We vary the exp and sin constants (c, t_1, t_2) and add Gaussian noise to the sampled points. In total, we generate 1000 example and use 80%-20% train-test split. For each example, we irregularly sample 200 points-showing the network a fraction of these points (i.e., the first 100 samples) and asking it to forecast the rest (i.e., the remaining 100 samples).

Figure 2 illustrates the prediction performance at different training stages with increased data and network complexity. We can see that by breaking down the complex curve into simpler curves, our PODE can learn complex functions incrementally. Figure 3 compares the forecasting performance between NODEs and PODEs. NODEs fail to represent the complex dynamics of our synthetic curves, yielding a mean squared error (MSE) of 15.56. In comparison, PODEs are capable of capturing both trend and seasonalities, producing an MSE Of 0.81, a substantial improvement. Table 1 compares the MSE of PODE against other baselines.

4.2 PeMS Bay Area Traffic Data

This dataset is collected by California Department of Transportation (Caltrans) using Caltrans Performance Measurement System (PeMS). We use the traffic flow readings - average vehicles on per unit time - aggregated in five minutes interval. This dataset incorporates both trend and seasonalities (including weekly and daily changes) and, therefore, is commonly used for time series forecasting. It also contains sufficient equally-spaced samples, which allow us to conduct experiments on the effect of irregular spacing with different sampling strategy (e.g., maximum/minimum spaces).

We randomly select one of the sensor data to be used, over a three year period from January, 2014 to December, 2017. Each sample in our data is a daily measurement of flow readings. This gives us 932 samples, after cleaning up days on which the sensor didn't collect any data, and 288 data points in each sample. As with synthetic data, we divide the dataset in 80%-20% train-test split ratio. We input half of 288 points to the network and ask it to forecast the latter half. Table 1 shows that PODE improves over NODE by more than 64%. Qualitative samples are visualized in fig. 4.

5 Conclusion

We proposed a novel progressive learning approach for modeling irregularly sampled time series data with complex trends and seasonalities. We demonstrated substantial improvements over state-of-the-art NODEs on both synthetic and real-world data. Our empirical study suggests that with the same architecture, network performance can be further improved by appropriate design of training procedure, such as curriculum learning, especially for complex tasks such as the forecasting of irregularly sampled time series with trends and seasonalities.

REFERENCES

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 41–48, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

George E. P. Box, Gwilym M. Jenkins, Reinsel Gregory C., and Greta M Ljung. *Time Series Analysis: Forecasting and Control*, volume 1. Wiley Series in Probability and Statistics, 2015.

Peter J. Brockwell and Richard A. Davis. *Introduction to Time Series and Forecasting*, volume 1. Springer, 2016.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations, 2018.

Jeffrey L. Elman. Learning and development in neural networks: the importance of starting small. *Cognition*, 48:71–99, 1993.

Everette S Gardner Jr. Exponential smoothing: The state of the art. *Journal of forecasting*, 4(1): 1–28, 1985.

Andrew C Harvey and PHJ Todd. Forecasting economic time series with structural and box-jenkins models: A case study. *Journal of Business & Economic Statistics*, 1(4):299–307, 1983.

Charles C. Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, 20:5–10, 2004.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation, 2017.

Steven Cheng-Xian Li and Benjamin M Marlin. A scalable end-to-end gaussian process adapter for irregularly sampled time series classification. In *Advances in neural information processing systems*, pp. 1804–1812, 2016.

Zachary C Lipton, David Kale, and Randall Wetzel. Directly modeling missing data in sequences with rnns: Improved classification of clinical time series. In Finale Doshi-Velez, Jim Fackler, David Kale, Byron Wallace, and Jenna Wiens (eds.), *Proceedings of the 1st Machine Learning for Healthcare Conference*, volume 56 of *Proceedings of Machine Learning Research*, pp. 253–270, Children's Hospital LA, Los Angeles, CA, USA, 18–19 Aug 2016. PMLR. URL http://proceedings.mlr.press/v56/Lipton16.html.

Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher-student curriculum learning. *CoRR*, abs/1707.00183, 2017. URL http://arxiv.org/abs/1707.00183.

Hongyuan Mei and Jason Eisner. The neural hawkes process: A neurally self-modulating multivariate point process, 2016.

Kira Rehfeld, Norbert Marwan, Jobst Heitzig, and Juergen Kurths. Comparison of correlation analysis techniques for irregularly sampled time series. *Nonlinear Processes in Geophysics*, 18:389–404, 2011.

Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. Latent odes for irregularly-sampled time series, 2019.

Jeffrey D Scargle. Studies in astronomical time series analysis. ii-statistical aspects of spectral analysis of unevenly spaced data. *The Astrophysical Journal*, 263:835–853, 1982.

Satya Narayan Shukla and Benjamin Marlin. Interpolation-prediction networks for irregularly sampled time series. 2018.

Wojciech Zaremba and Ilya Sutskever. Learning to execute. *CoRR*, abs/1410.4615, 2014. URL `http://arxiv.org/abs/1410.4615`.

G Peter Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.