Few-shot Time-Series Forecasting with Known Information using Black-Box Optimization

Arec Jamgochian (arec@), Sheng Li (lisheng@) and Xiaobai Ma (maxiaoba@)

Abstract

Accurate time-series forecasting is important for many applications. Well-studied examples include weather, electricity consumption, traffic, stock market, and sales forecasting. The typical forecasting problem is to use a long sequence of observations $\mathbf{y_{1:H}}$ to learn a model that predicts a next-step observation given a shorter sequence of observations, $y_{t+1} \mid \mathbf{y_{t-L+1:t}}$. Many forecasts can be further improved upon by providing external information $\mathbf{e_t}$ that is known at inference time. Examples of useful external information include time-of-day if the time-series exhibits daily periodicity, or temperature if weather could play a latent role in the time-series observations. Learning a model that can accurately infer $y_{t+1} \mid \mathbf{y_{t-L+1:t}}, \mathbf{e_t}$ for a single time-series typically requires a lot of data for that time-series.

In this paper, we explore time-series forecasting when given short time-series in our target task, but longer time-series in many related tasks. Meta-learning approaches seek to alleviate issues with insufficient data by leveraging data from different but related time-series. With black-box optimization methods, one forms a single model that takes as input short support and query data and directly outputs predictions, training this model using the many related time-series. In memory-augmented neural networks (MANN), recurrence is performed over support data and query inputs to learn useful hidden states for prediction [Santoro et al., 2016]. We first formulate the time-series forecasting problem using MANN with appended information inputs to help with forecasting (InfoMANN). We next contribute adjustments such that recurrence at prediction time is performed over time in the query sequences rather than over the sequences themselves (TSMANN). TSMANN uses the final hidden state from support to initialize the recurrent network at prediction time, thereby encoding a 'rulebook' to use for inference. TSMANN also allows us to make predictions about an arbitrary number of queries in parallel, which is more useful to time-series forecasting, while the traditional MANN structure requires recurrence over a fixed number of prediction queries.

Our experiments focus on forecasting hourly residential electricity consumption when the support time-series are short. This is useful to utility companies who wish to build forecasting models at new smart meter installation locations. We test our methods on the OPENEI TMY3 residential dataset [OPENEI, 2020], focusing on building a model to make predictions when given queries of $L = 8$ past observations, embeddings consisting of daily, weekly, and yearly period information, and $H = 504$ indices (three weeks) of support data[1]. We find that both InfoMANN and TSMANN exhibit lower mean squared error at test time compared to models trained on each support set individually. We find that TSMANN does indeed improve upon InfoMANN, which in turn improves upon MANN. We do however see that on this particular dataset, combining data across time-series into an augmented dataset and learning a single prediction model for $y_{t+1} \mid \mathbf{y_{t-L+1:t}}, \mathbf{e_t}$ regardless of time-series does in fact outperform all MANN-based models. We posit that this is due to similar trends across houses in the dataset. We hope to address this in future work.

[1] Code available at https://github.com/jamgochiana/MetaProbabilisticLoadForecasting (requires permission from authors).

1 Introduction

In this project, we study the applications of meta-learning to few-shot forecasting of low-dimensional time-series. Accurate time-series forecasting has use in many applications. For example, as a society we are highly dependent on forecasts of weather, markets, and traffic. In time-series forecasting, the goal is to make predictions about future observations give a stream of H observations $\mathbf{y_{1:H}}$. Typically, rather than building an inference model using the whole data stream, it is more useful to split the input stream into shorter sequences of length L and build an inference model on the shorter sequences – $p(y_{t+1}|\mathbf{y_{t-L+1:t}})$. Doing so allows you to leverage your data to build an inference model that better captures short-term dependencies, at the expense of long-term correlations.

We can improve forecasts by leveraging external information that is known a priori about possible latent features. For example, many time-series experience periodicity–we can expect different observations at different periods of time (e.g. daily cycles in traffic, yearly cycles in weather). As another example, you can expect electricity consumption to be highly correlated with weather. Is is therefore often useful to encode this external information $\mathbf{e_t}$, and to learn a joint inference model $p(y_{t+1}|\mathbf{y_{t-L+1:t}}, \mathbf{e_t})$.

Learning a good inference model requires a sufficient amount of data. However, we may not always have sufficient data from our target time-series, but may have sufficient data from other related time-series. In meta-learning, we wish to learn how to efficiently use a small amount of data (the *support* set) to learn a predictive model (to be used on a *prediction* set), given a lot of related data (a lot of related support and prediction sets). In this paper, we take the black-box optimization approach to meta-learning – that is to use all the related data to learn a single model that can take in a stream of support data and make good predictions on the associated prediction data. Specifically, we focus on adapting Memory-Augmented Neural Networks [Santoro et al., 2016] (MANN) for use with time-series.

In our experiments, we will focus on forecasting of electricity consumption. Electricity consumption is fundamentally stochastic and influenced by unseen latent variables (e.g. weather), however accurate load forecasting is critical to making dynamic resource allocations in the electricity grid. The few-shot load forecasting setting is useful when not a lot of data is present–for example, when a new installation of a smart meter is made. A sample of five days of residential electricity consumption data can be seen in Fig. 3.

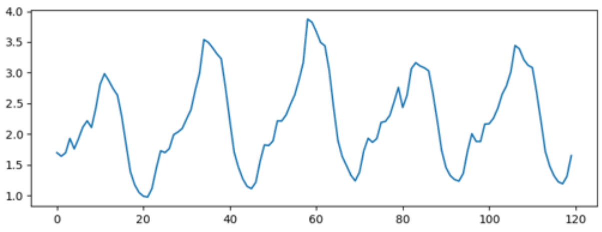

Figure 1: Sample hourly residential electricity consumption from the OPENEI dataset [OPENEI, 2020], showing daily periodicity.

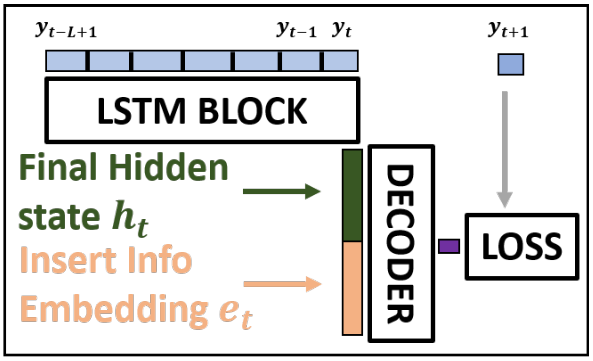

Figure 2: A state-of-the-art architecture for electricity forecasting with information augmentation [Wang et al., 2019]. Recurrence is performed over time, and information embeddings are appended to the final hidden state before decoding.

One popular model for characterizing $p(y_{t+1}|\mathbf{y_{t-L+1:t}}, \mathbf{e_t})$ uses a recurrent neural network to perform recurrence over time in the target data stream, then appends the information vector with the final hidden state before decoding [Wang et al., 2019]. This architecture, which can be seen in Fig. 2, unfortunately requires a lot of target series data. Though MANN requires little target data, it traditionally performs recurrence over each element in the support and prediction set, rather than over time.

Our contributions in this project are to:

- extend memory-augmented neural networks for use for few-shot information-augmented forecasting (InfoMANN),
- improve on InfoMANN by performing recurrence over time during prediction (TSMANN), and
- test our methods on an electricity consumption dataset.

Section 1.1 details related work, Section 2 details our methods, Section 3 details our experiments, and we conclude in Section 4.

1.1 Related Work

In model-agnostic meta-learning [Finn et al., 2017], parameters are learned that enable quick fine-tuning on the meta-test task. With memory-augmented neural networks (MANN) on the other hand, a model is maintained that learns the entire few-shot training process [Santoro et al., 2016].

There are existing works in literature about multi-task time-series forecasting. Jin and Sun [2008] use multi-task learning to take advantage of the information provided by related tasks and to improve generalization by transferring information in training signals of extra tasks. Multi-task learning is used for traffic flow forecasting. Fiot and Dinuzzo [2016] introduce kernel-based multi-task learning techniques to forecast the demand of electricity measured on multiple lines of a distribution network. Their approach allows to flexibly model the complex seasonal effects that characterize electricity demand data, while learning and exploiting correlations between multiple demand profiles. Cirstea et al. [2018] use a combination of convolutional neural network, auto-encoder and recurrent neural network to achieve multi-task learning for forecasting in correlated time-series of cyber-physical systems. Fan et al. [2019] propose a novel end-to-end data-driven approach for solving multi-horizon probabilistic forecasting tasks that predicts the full distribution of a time-series on future horizons. Temporal attention mechanism is used to better capture latent patterns in historical data which are useful in predicting the future. Oreshkin et al. [2020] investigate multi-task learning for time-series in the related zero-shot setting, reporting results on the UCI Electricity dataset (among other things). To our knowledge, no work explicitly considers few-shot probabilistic electricity load forecasting.

MANN has also been used for time-series forecasting. Li et al. [2020] use MANN to enhance the demand prediction of knowledge-sparse public transportation modes with the data from knowledge-intensive modes by deriving the transferable demand patterns from each mode and boost the prediction of knowledge-sparse modes through adapting the relevant patterns from the knowledge-intensive modes. Marchetti et al. [2020] propose a MANN based model that exploits memory augmented networks to effectively predict multiple trajectories of other agents, observed from an egocentric perspective. The model stores observations in memory and uses trained controllers to write meaningful pattern encodings and read trajectories that are most likely to occur in future.

There has been growing recent interest in probabilistic load forecasting [Hong and Fan, 2016]. Wang et al. [2019] achieve state-of-the-art probabilistic performance using an LSTM with additional seasonality embeddings in order to forecast quantiles of a predictive next-step distribution. They append the information to a recurrent network hidden state embedding warmed-up by the input time-series.

2 Methods

In this section we outline:

1. how to sample sub-sequences from longer time-series for use in black-box meta-training,

2. information-augmented MANN, which performs recurrence over sequences to learn a useful meta-model, and

3. time-series MANN, which performs recurrence over time in the prediction sequences.

2.1 Data Generation Process

Recall that the goal of this project is to transfer knowledge gained from other longer time-series to learn a good model given a shorter (support) time-series. Our assumption is that observations y within each time-series (series s) in our meta set are drawn consistently from a single distribution $p_s(y_{t+1}|\mathbf{y_{t-L+1:t}}, \mathbf{e_t})$ for that time-series, where L is a short lag index and $\mathbf{e_t}$ is an information embedding vector. During black-box meta-training, we must therefore learn to characterize p_s with little data from p_s, but a lot of data from $p_{\setminus s}$.

We assume that we will have H time-steps of support data in our target problem. Our resulting data generation process is highlighted in Fig. 3, and can be summarized in the following steps:

1. Sample a meta-batch of B random time-series.

2. For each time-series, sample H indices for support and save the subsequent H indices for prediction.

3. From each full support sequence, extract every $(\mathbf{y_{t-L+1:t}}, \mathbf{e_t}, y_{t+1})$ tuple for support.

4. From each prediction sequence, extract T random $(\mathbf{y_{t-L+1:t}}, \mathbf{e_t}, y_{t+1})$ tuples for prediction.

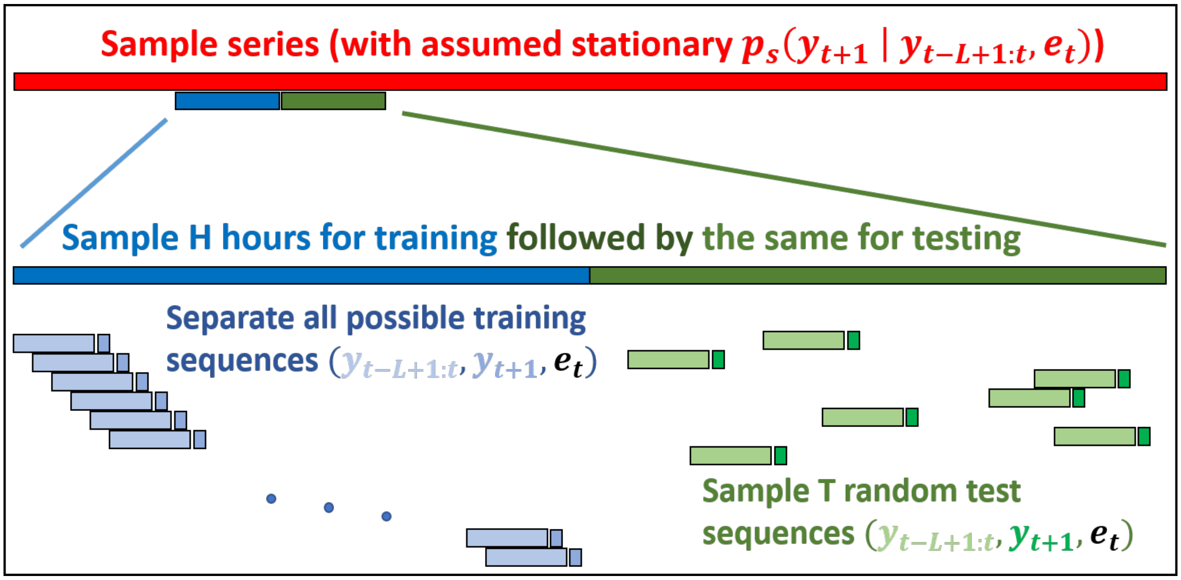
Sample series (with assumed stationary $p_s(y_{t+1} \mid y_{t-L+1:t}, e_t)$)
Sample H hours for training followed by the same for testing
Separate all possible training sequences $(y_{t-L+1:t}, y_{t+1}, e_t)$
Sample T random test sequences $(y_{t-L+1:t}, y_{t+1}, e_t)$

Figure 3: Our time-series data generation procedure (information embedding extraction not shown).

2.2 Information-Augmented MANN

Once we generated all training and testing tuples, we may run our tuples through a information-augmented memory-augmented neural network (InfoMANN), depicted in Fig. 4. InfoMANN augments MANN with the information embeddings. Each support and prediction sequence is treated as an individual data vector, and recurrence is done over all data vectors to learn an encoding for 'rules'. As with MANN, the L-step sequences are appended to the next-step labels as input during the support section, while next-step labels are nullified in the prediction inputs and used instead in the loss function.

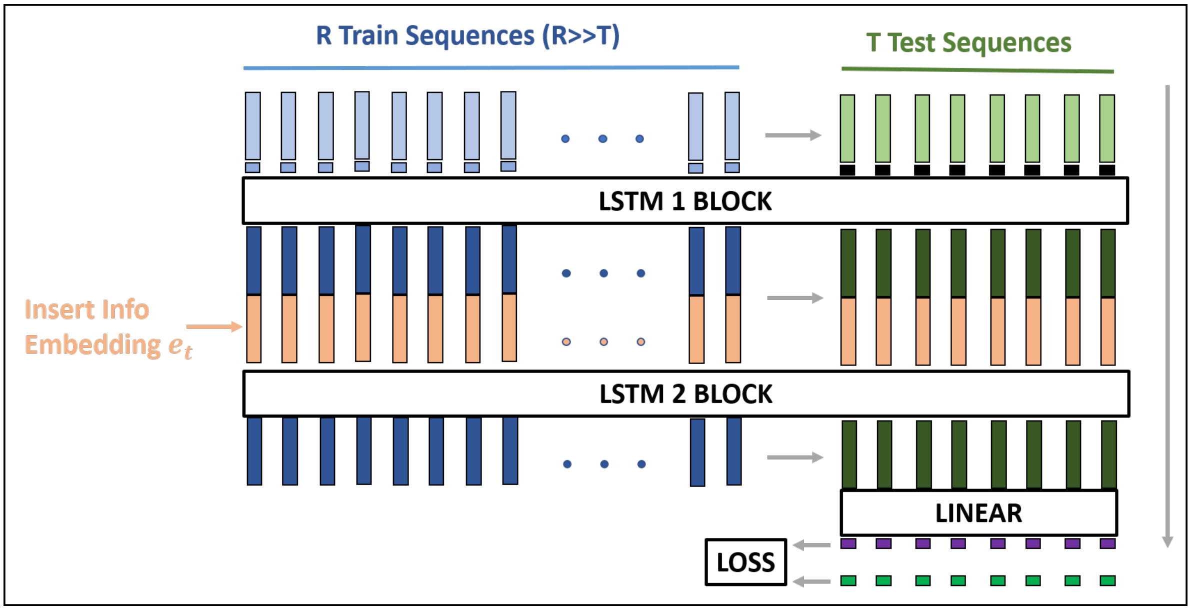

Figure 4: Information-augmented memory-augmented neural network (InfoMANN) with information embeddings inserted after the first LSTM layer. The L-step sequences are appended to the next-step labels as input during the support section, while next-step labels are hidden in the prediction section.

Typically, one might be interested in learning to make good point forecasts, and so therefore might want to use mean-squared error as a loss function to be minimized. However, our meta-learning methods are also amenable to probabilistic forecasting, the difference being the length of the prediction vector (purple in the figures) and the loss function. A table of popular forecasting schemes may be found below, with Θ_i being the prediction vector for the i-th prediction query, and $|\Theta_i|$ being the required cardinality of the meta-model's output vector. Note that additional measures need to be taken to properly specify the probability distributions (e.g. standard deviations should be positive, weights should be positive and sum to one, α-quantiles should be non-decreasing).

Table 1: Popular schemes for single-step forecasting, and their implications for use in these black-box optimization architectures.

Setting	**Description**	Θ_i	$\|\Theta_i\|$	**Loss**
Point	Vanilla Point	μ_i	1	MSE
	Robust Point	μ_i	1	Huber [Huber, 1964]
Prob.	Gaussian	μ_i, σ_i	2	NLL
	GMM	$\{w_i^k, \mu_i^k, \sigma_i^k\}_{k=1}^K$	3K	NLL
	Quantiles	$\{\alpha_i^q\}_{q=1}^Q$	Q	Pinball [Koenker and Bassett Jr, 1978]

2.3 Time-Series MANN

InfoMANN has some drawbacks. First and foremost, we are only limited to reliably using InfoMANN when we draw T random prediction query sequences from the future, which is impractical for time-series forecasting. You generally only have one query sequence at a time – the past L inputs. We may try to set $T = 1$ and iteratively feed the past L inputs to guess the next one, but we cannot practically use this model to make predictions at T sequences (unlike for example, in the K-shot, N-way classification setting) because $T-1$ of those query sequences are unknown.

Another drawback is that this system does not leverage the sequential nature of the problem to make predictions. It treats all sequences as individual data vectors, and the recurrence is done over each input vector. In contrast, state of the art methods for time-series forecasting (e.g. Fig. 2) perform recurrence over time, and then append information embeddings to extract useful predictions. These methods exist in the non-meta learning framework–the goal of the recurrence is to learn how to interpret and store useful information from the time-series, that is learned in a way designed to be useful to inputs regardless of their source.

We propose to address both of these concerns by implementing a sort of 'conditional' prediction recurrent network, that performs recurrence in sequences over time conditioned on a set of 'rules' that are learned from support. In time-series MANN (TSMANN), we adjust the previous architecture such that after all support sequences are seen, the final hidden state is used to initialize LSTMs that perform recurrence over time for prediction sequences. See Fig. 5 for an illustration.

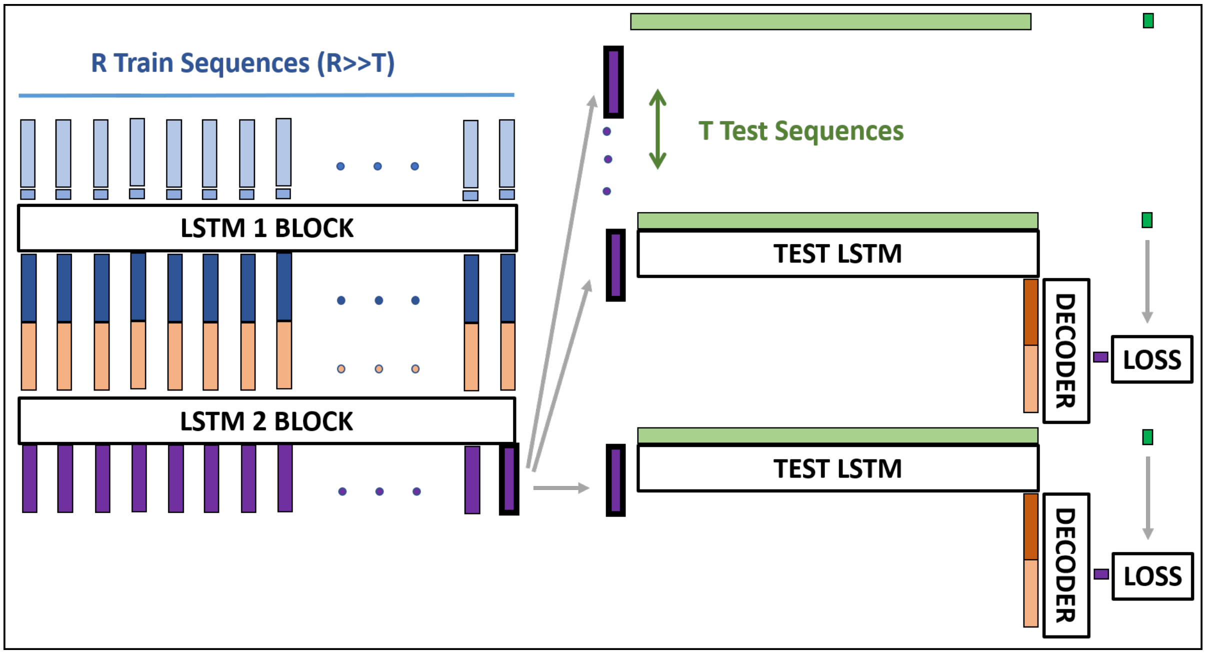

Figure 5: Memory-augmented neural network for time-series with information augmentation (TSMANN). Recurrence is done over all elements in the support set. The final hidden encoding is used as an initialization scheme for prediction networks, which perform recurrence over time and leverage information during decoding, as is done in the presence of lots of data [Wang et al., 2019].

To learn a model, we may use the final support hidden state as an initial hidden/cell state for T identical (shared parameter) LSTMs, which then apply time recurrence over the T test sequences to form final hidden states that get used for forecasting next step in each sequence. The goal of meta-learning in this framework is fewfold: a) to learn a good 'set of rules' (final hidden state) from support that can be used during prediction and b) to learn a good prediction model that can interpret those 'rules' when taking in a test sequence to learn a useful hidden state for prediction.

Once trained, this model can then be used to test an arbitrary number of sequences, simply by initializing the starting hidden states for each sequence using the final hidden state from the support section.

3 Experiments

In our experiments, we aim to empirically prove the hypothesis that using meta-learning can indeed help learn a good forecast model for a time-series with insufficient data. We focus on building meta-trained models for electricity consumption forecasting. This application has real-life benefits – better electricity consumption forecasting with scarce data would give utility companies models to use at locations with recent smart meter installations.

3.1 Setup

Dataset We use the OpenEI TMY3 dataset [OPENEI, 2020], which reports hourly consumption patterns across different residential and commercial locations at different locations across the United States. We test our model with the 973 residential locations, which we split into 80/10/10% train/val/test series. Five days of residential consumption at a sampled location are shown in Fig. 1.

Problem Setting For our goal, we aim to build a good model for predicting next-hour load given the past $L = 8$ hours of load. To capture seasonal, weekly, and hourly trends, we form information embeddings of cyclically encoded hour-of-day, day-of-week, and month-of-year. We focus on building these models given 3 weeks ($H = 504, R = 496$) of training data and $T = 20$ randomly sampled test sequences. We also focus on point forecasting, hoping to minimize mean-squared error of forecasts on our test dataset.

MANN Methods We compare two MANN approaches as introduced in Sec. 2.

1. InfoMANN: Following the structure described in Sec. 2.2, the InfoMANN inserts information embeddings at the input layer and uses a hidden dimension of 16 for both LSTM blocks.

2. TSMANN: Following the structure described in Sec. 2.3, the TSMANN uses a hidden dimension of 16 for all 3 LSTM blocks and the decoder uses two hidden layers with 16 hidden units.

Baselines To test the usefulness of meta-learning, we also implement two baselines:

1. MLP Per Task: An MLP model that is trained separately per task given only the 3 weeks of training data. The MLP model takes the concatenation of loads from the past $L = 8$ hours as well as the info embedding and output the predicted next hour load. It uses 5 hidden layers with 32 hidden units to match the parameter number used in MANN models.

2. MLP All: An MLP model that is trained globally using an aggregation of all time-series in the meta-training set. This model takes in queries while regarding their source time-series. The network structure is the same as in MLP Per Task.

3.2 Results

Prior to running method comparisons, we perform some basic architecture design and compare where information should be inserted into InfoMANN, if at all. The results (Fig. 6) show clear improvements when using information embeddings, and show similar improvements when inserting embeddings at the input layer or after the first layer. This justifies the use of InfoMANN over MANN.

Next, to compare the methods, each method is trained with 2×10^4 iterations using a batch size of 32 and a learning rate of 1×10^{-3}. Each training is repeated with 3 different random seeds. The mean as well as the standard deviation of the validation mean squared errors (MSEs) are plotted versus training iteration in Fig. 7. Since the amount of the training data is significantly less in the MLP Per Task, there is a severe overfitting. Thus, we train MLP Per Task with 2×10^3 iterations and report the minimum MSE across the training averaged over all tasks in the meta-training/validation set.

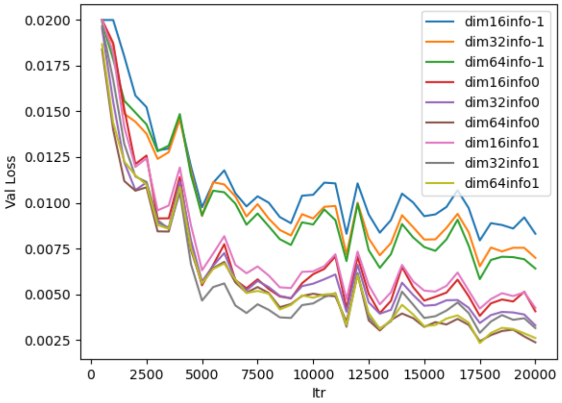

Figure 6: InfoMANN architecture design, comparing validation mean squared error when using different hidden dimensions (16, 32, 64) and when inserting information embeddings at different layers (not at all, layer 0, layer 1).

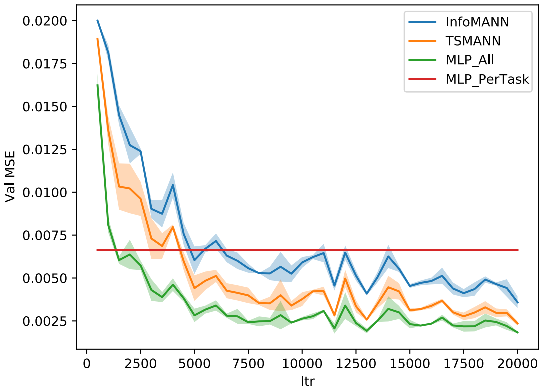

Figure 7: Comparison of InfoMANN, TSMANN, and baselines. The horizontal red line indicates the average of the minimum MSE from the MLP Per Task baseline.

The training MSE of MLP Per Task is 0.0013, less than the two MANN experiments. This is expected since the MLP Per Task model easily overfits the training data in each task. However, the performance of MANNs on the meta-validation set is significantly better than MLP Per Task as shown in Fig. 7. This validates the MANN approaches' ability to aggregate useful information in the meta-training data and generalize it on the meta-validation data. We also notice that the TSMANN behaves consistently better than InfoMANN on both meta-training and meta-validation, which verifies our hypothesis that TSMANN should behave better than InfoMANN by leveraging the sequential nature of the prediction problem. The performance of the MLP All is unexpectedly better than MANN approaches. Since MLP All is trained solely on the meta-training data without seeing any training data in meta-validation set, this result suggests that the distribution difference across meta-train and meta-validation tasks in the dataset is not sufficiently large. We hope to address this in future work.

4 Conclusion

In this project, we studied the application of meta-learning methods on time-series prediction tasks. Specifically, we explored two variants of the memory-augmented neural network (MANN) approaches, namely InfoMANN and TSMANN, on the OPENEI TMY3 residential dataset. In InfoMANN, we studied the affect of augmenting the model with information embeddings to better capture external but known features (such as periodicity in the time-series). Further in TSMANN, we demonstrated how a LSTM encoder could be trained to embed a good 'set of rules' in its final hidden state from the training data and applied it with a LSTM decoder on different series in the test data. Our experiments show that both InfoMANN and TSMANN are able to get better performance than the MLP baseline trained only using the per task data. For future work, we would like to investigate the performance difference between MANN methods and supervised learning baselines as we increase the difference between individual tasks. We would also like to explore the application of model agnostic meta learning (MAML) approaches on time-series predictions.

References

Razvan-Gabriel Cirstea, Darius-Valer Micu, Gabriel-Marcel Muresan, Chenjuan Guo, and Bin Yang. Correlated time series forecasting using multi-task deep neural networks. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 1527–1530, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360142. doi: 10.1145/3269206.3269310. URL https://doi.org/10.1145/3269206.3269310.

Chenyou Fan, Yuze Zhang, Yi Pan, Xiaoyue Li, Chi Zhang, Rong Yuan, Di Wu, Wensheng Wang, Jian Pei, and Heng Huang. Multi-horizon time series forecasting with temporal attention learning. page 2527–2535, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330662. URL https://doi.org/10.1145/3292500.3330662.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

Jean-Baptiste Fiot and Francesco Dinuzzo. Electricity demand forecasting by multi-task learning. *IEEE Transactions on Smart Grid*, 9(2):544–551, 2016.

Tao Hong and Shu Fan. Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting*, 32(3):914–938, 2016.

Peter J Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, pages 73–101, 1964.

Feng Jin and Shiliang Sun. Neural network multitask learning for traffic flow forecasting. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1897–1901. IEEE, 2008.

Roger Koenker and Gilbert Bassett Jr. Regression quantiles. *Econometrica: journal of the Econometric Society*, pages 33–50, 1978.

Can Li, Lei Bai, Wei Liu, Lina Yao, and S. Travis Waller. Knowledge adaption for demand prediction based on multi-task memory neural network. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 715–724, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368599. doi: 10.1145/3340531.3411965. URL https://doi.org/10.1145/3340531.3411965.

Francesco Marchetti, Federico Becattini, Lorenzo Seidenari, and Alberto Del Bimbo. Multiple trajectory prediction of moving agents with memory augmented networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

OPENEI. Available [online]http://en.openei.org/doe-opendata/dataset. 2020.

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. Meta-learning framework with applications to zero-shot time-series forecasting. *arXiv preprint arXiv:2002.02887*, 2020.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850, 2016.

Yi Wang, Dahua Gan, Mingyang Sun, Ning Zhang, Zongxiang Lu, and Chongqing Kang. Probabilistic individual load forecasting using pinball loss guided lstm. *Applied Energy*, 235:10–20, 2019.