Stacked Boosters Network Architecture for Short Term Load Forecasting in Buildings

Tuukka Salmi, Jussi Kiljander, Daniel Pakkala
VTT Technical Research Centre of Finland
Oulu, Finland
tuukka.salmi@vtt.fi, jussi.kiljander@vtt.fi, daniel.pakkala@vtt.fi

Abstract—This paper presents a novel deep learning architecture for short term load forecasting of building energy loads. The architecture is based on a simple base learner and multiple boosting systems that are modelled as a single deep neural network. The architecture transforms the original multivariate time series into multiple cascading univariate time series. Together with sparse interactions, parameter sharing and equivariant representations, this approach makes it possible to combat against overfitting while still achieving good presentation power with a deep network architecture. The architecture is evaluated in several short-term load forecasting tasks with energy data from an office building in Finland. The proposed architecture outperforms state-of-the-art load forecasting model in all the tasks.

Index Terms—Deep neural networks, Short term load forecasting

I. INTRODUCTION

Due to increasing utilization of renewables controlling the demand flexibility is becoming crucial part of the stabilization of smart grids. In this setting individual buildings are becoming key resources since buildings consume 32% of global final energy use [1]. Fundamental part of controlling demand flexibility of individual buildings is accurate building level short term load forecasting.

Artificial neural networks (ANN) provide a rich machine learning method family for time series forecasting problems. According to a recent study by Amasyali et al. [2], ANNs are the most commonly used machine learning model in short term forecasting of building energy loads. The challenge with artificial neural networks is that they require a lot of data to prevent overfitting. This is usually tackled by reducing the number of layers, which in turn reduces the presentation power of the model. For instance, a very typical neural network architecture in building load forecasting is a single hidden layer multilayer perceptron (MLP) model [3], [4], [5], [6]. Besides neural network methods, support vector regression (SVR) is a common machine learning model for building load forecasting [7], [8], [9]. Although SVR is not a neural network model, it can be though as a single hidden layer ANN from the point view of the model capacity.

In addition to the shallow MLP and SVR models, deeper model architectures for building load forecasting have been proposed. Marino et al. [10] compare standard long short term memory (LSTM) and LSTM sequence-to-sequence architectures (also known as encode-decoder) for building energy load forecasting. LSTM based architecture is also studied by Kong et al. in two papers [11], [12]. Although the LSTM architectures studied in all three papers consist of only two stacked LSTM layers, the recurrent neural networks (RNN) such as LSTM are deep models due to the recursive call made at every time step. Amarasinghe et al. [13] present a CNN-LSTM network consisting of two CNN layers and single LSTM layer. The model is evaluated in five different building datasets. Mocanu et al. [15] study the performance of conditional restricted Boltzmann machine (CRBM) and factored conditional restricted Boltzmann machine (FCRBM) neural network architectures in short-term load forecasting.

A very deep ANN architecture for energy load forecasting is presented by Chen et al. [16]. The authors adopt residual connections [17], a successful approach for building deep CNNs for image processing, and form a 60 layer deep network for energy load forecasting. The proposed ResNet borrows some ideas from RNN type network since it contains connections from previous hour forecast to next hour forecasts, but the connection is not an actual feedback since there are no parameter sharing among different hour forecasters. The proposed ResNet combined with an ensemble approach achieves state of the art results in three public energy forecasting benchmarks. Although none of these datasets focuses on individual building loads, the proposed work is general energy load forecasting model and thus good state-of-the-art benchmark also for building level forecasting tasks.

The aforementioned deep learning models have been evaluated using datasets with large amount of relatively static training data, which allows the models to avoid overfitting and outperform shallower models. However, in many situations (e.g. with new buildings or locations with extreme weather conditions) it is useful to be able to avoid overfitting also in situations with limited amount of training data.

We propose a novel hierarchical neural network architecture for short term load forecasting. The architecture, called Stacked Booster Network (SBN), tries to achieve the good...
properties of deep models while still avoiding overfitting in small sample size situations. The core idea is to reduce the model parameter space with following principles: 1) sparse interactions, 2) parameter sharing, and 3) equivariant representations. Another key idea of the architecture is a novel boosting technique, which makes it possible to transform the original multivariate time series problem into univariate one. This further reduces the model parameters while keeping the network capacity high enough. Additionally, the proposed boosting technique enables the model to correct systematic mistakes by utilizing residual information on historical forecasts. With these ideas we can build a deep learning framework for short term load forecasting with following properties:

- Minimal number of parameters leading to robust training even with small amount of training data
- Sufficiently large number of layers leading to enough presentation power for modelling real phenomenon of the modelled load
- Boosting technique that allows the model to adapt to changing data distributions, which are typical in real-life data

The paper is organized as follows. In section II, general network architecture of SBN is presented. Section III introduces the case studies and presents the evaluation where SBN is compared to the state-of-the-art ResNet model [16]. Section IV concludes the presentation and presents directions for future work.

II. SBN ARCHITECTURE

General model architecture is described in Figure 1. The architecture is composed of four level forecasters:

- Instant forecaster
- Hourly boosting forecaster
- Daily boosting forecaster
- Weekly boosting forecaster

Forecasters are designed to be stacked over previous forecaster in a way that the next forecaster boosts the previous one. This way every boosting forecaster performs only univariate time series modelling although the original time series would be multivariate. Only Instant forecaster handles the original multivariate time series. The architecture of the whole boosting system is shown in Figure 1 while the architecture of an individual booster is shown Figure 2. Architecture of Instant forecaster is shown Figure 3.

Next subsections cover each of the forecasters with more details with their limitations.

A. Instant forecaster

The purpose of Instant forecaster is to forecast energy consumption based on independent variables that are assumed to be fully observable without using any information on past energy consumptions. In our case study that is described with details in Section III the only independent time series are temperature, hour of the day, and day of the week. Here also temperature can be considered fully observable by replacing
future temperature values by their forecasts. These independent variables are first feed with dimensionality reduction submodels to reduce time dimension of these univariate time series to one. Considering our case study, hour of the day and day of the week are deterministic time series and they need not to use previous values at all and therefore they do not need dimensionality reduction submodel.

It should be noted that we make a reasonable assumption that different temperature profiles behaves exactly same way at every weekday and every hour. Therefore, with this assumption reducing temperature sequence dimension to one before merging to other inputs does not lose any usable information. If some other information sources would be used, it is crucial to analyse whether the approach loses information and whether extra complexity needs to be addressed in the submodel.

B. Weekly boosting forecaster

Energy time series have typically periodic loads that occur same time on specific days of the week. If the loads would be always at the same level at the same time, Instant forecaster would estimate them perfectly. However, it is quite typical that the timing and volume of these loads varies over time. This is the phenomena that Weekly boosting forecaster should compensate. Consider, for example the load occurring at some specific weekday rapidly increases by fixed amount. Then Weekly boosting forecaster gets time series having this constant value as an input, and forecasts this constant value for the future error also. Therefore, in this case Weekly boosting forecaster is capable for estimating this dynamic change perfectly after short reaction time.

C. Daily boosting forecaster

Daily boosting forecaster works exactly same way as Weekly boosting forecaster but it focuses on changes on loads that occurs daily bases rather than weekly basis. Consider, for example that normally daily basis occurring load rapidly increases by fixed amount. Weekly boosting forecaster would need weeks data to be able to estimate and correct it. Therefore, it remains uncorrected at the beginning. Therefore daily boosting forecaster gets error time series with constant value as input, and forecasts this constant value for the future error also.

D. Hourly boosting forecaster

Some energy time series have periodic fluctuations in the energy consumption that cannot be explained purely by the input data. This can be explained by phenomena where some devices may be turned on and off periodically and this periodicity can be changing by some unknown conditions. Therefore, this phenomena can be seen as error signal of Instant forecaster. Moreover, if the periodicity is not divisible by 24 hours, Weekly boosting forecaster and Daily boosting forecaster cannot compensate it. This is the phenomena that Hourly boosting forecaster should compensate. Another more minor phenomena that hourly boosting forecaster should compensate is if some constantly occurring load get changed. In this case weekly and daily boosting forecasters react with days delays and Hourly boosting forecaster can react and estimate the errors more rapidly.

E. Implementation notes

It should be noted that stacking of Hourly boosting forecaster, Daily boosting forecaster and Weekly boosting forecaster can be done in any order. Moreover, some of the forecaster may not be needed at all. In fact, higher time resolution boosting forecaster can correct exactly the same issues that lower time resolution forecaster when the time window it gets as input is big enough. However, due to fact that some energy load occurs typically in daily and weekly basis, these energy loads are more easily estimated and compensated by Daily boosting forecaster and Weekly boosting forecaster. Our choice is to stack boosters in order Weekly boosting forecaster, Daily boosting forecaster and lastly Hourly boosting forecaster since in this order boosters are not capable to correct such phenomenons that following booster can do it better. However, we have not experimented different ordering the boosters and our experiments presented in Section III uses only this fixed stacking order.

One nice feature of the architecture is that it splits the big problem in small separate problems that can be optimized separately when stacking the forecasters. First the network performing Instant forecaster should be implemented and optimized. This network architecture can be optimized without need of integrating the boosting forecasters. Then, the first boosting forecaster is implemented and stacked on top of Instant forecaster. The network architecture of this boosting forecaster can now be optimized separately. Similarly all the network architectures of the other booster forecasters can be optimized one by one when stacking those. It is still quite open issue whether this iterative approach will provide an optimal total network architecture. The iterative approach seems to offer reasonable good architecture with easy design flow.

However, a finding is that this iteratively approach is not always optimal what it comes to training the network weights. This issue is covered in the next section.

F. Training the network

The proposed architecture offers multiple different ways of training the network.

- Train the whole network at once using the final prediction value only in loss function
- Train the whole network at once using all the forecasters outputs in loss function but possible with different weights.
- Train first the first prediction submodel and freeze it. Then, train add one boosting forecasters one by one freezing all the earlier forecasters.
- Train iteratively as in previous bullet but do not freeze the earlier layers.

We have evaluated all the approaches. The second and the fourth provided equal good results while the second one is naturally faster since it needs only one training round. It seems
that providing some weights to earlier forecaster loss functions fights well against overfitting and it can be considered as a kind of regularization technique. This technique can also be considered as a kind of a short-cut connection similarly as in Resnets [17], [19].

It seems that the training is not very sensitive to weight given to Instant forecaster and earlier boosting forecasters. In our case study, we used weight 0.1 for Instant forecaster and earlier boosters and weight 0.9 for the final booster.

III. CASE STUDY

SBN architecture is applied for forecasting thermal energy consumption of Finnish office building.

The main problem is to forecast hourly thermal energy consumption of the building for 24h in advance. In addition to this main problem, we evaluate the performance in 48h and 96h forecasts also. Moreover, the effect of the training data size is evaluated by altering the training data size from 6 months to 6 years. The target metric is RMSE.

Available data are the past hourly energy consumption and temperature measurements. Temperature measurements are measured from a weather station being few kilometres away from the building. For the future timestamps, temperature forecasts are not used in this experiment but real temperature readings from the future timestamps.

A. Dataset

Thermal energy dataset contains hourly energy measurements and hourly temperature measurements from 1.2.2012 to 31.12.2018.

B. Methodology

The model architecture was optimized by using years 2012-2016 for training and year 2017 for the validation using manual architecture search. When reasonable model architecture was found, the final performance metric was run from year 2018. All the runs that was executed for year 2018 are presented in this paper. After running the metrics for year 2018 no feedback to model parameters was made anymore.

C. Model

1) Instant forecaster: Instant forecaster has three inputs:
1) Twelve consecutive hours temperature readings before each forecast hour are used.
2) Dummy encoded vector of length 2 determining whether the day is Saturday, Sunday or regular weekday.
3) Predicted hour. Hour is encoded onto unit circle to get continuous variable. Hour is therefore two length vector.

It should be noted that there are many instances of Instant forecast submodel, and all the instances shall share the same parameters. Twelve temperature values are first fed through dimensionality reduction submodel that reduces the temperature dimension to one and then all the inputs are processed by two layer densely connected network to get simple forecasts where the hidden layer contains 32 units with dropout regularization.

2) Boosting forecasters: The problem shall be solved by stacking all the three boosting forecasters with different combinations. For simplicity all the boosting forecasters shall use the same structure having two fully connected layers where the hidden layer contains 32 units with dropout regularization. Weekly boosting forecaster contains 3 weeks data. Therefore, we have values $N = 3$ and $k = 1$ in Figure 2. Daily boosting forecaster contains 7 days data indicating values $N = 7$ and $k = 1$ for 24 hour forecast in Figure 2. Finally, Hourly boosting forecaster contains 24h data, indicating values $N = 24$ and $k = 24$ for 24h forecast. It should be noted that there are multiple instances of each boosting forecaster submodels and the submodels shall share the same weights. However, for different boosters, submodel weights are naturally different.

Some statistics of network architectures for each of the used booster setups is shown in Table I.

D. Performance comparison

Performance comparisons of the usage of the different booster setups are presented in Table II and Table III. First, the Table II compares the setups by altering the training data size. Here one should note that basic Instant forecaster is the most sensitive to the training data size. When the training data size is only 6 months, and there are no training data available from all the temperature conditions, the performance of Instant forecaster decreases a lot. Nevertheless, boosting forecasts are still available to correct these errors to some extent. Moreover, when the training data size increases to 6 years, aging of the data starts to decrease performance of Instant forecaster. Again the boosting forecasters can compensate these errors to some extent. From Table III one should note that usage of Hourly boosting forecaster gets useless when the forecasting period increases since old hourly energy consumptions do not provide any correlation to future data of different hours. It eventually decreases performance due to overfitting.

1) Performance comparison to the state-of-art: For performance comparison to state-of-the-art we used Residual network approach [16] as presentation of state-of-the-art since

| Method       | Num layers | Num parameters |
|--------------|------------|----------------|
| Instant      | 3          | 238            |
| Daily        | 5          | 527            |
| Weekly       | 5          | 399            |
| Weekly and daily | 7          | 624            |
| Weekly, daily and hourly | 9         | 1457           |

Table I. Architectures of Different Booster Combinations

| Method/training data size | 6 months | 1 year | 6 years |
|---------------------------|----------|--------|---------|
| Instant                   | 6.28     | 3.64   | 4.46    |
| Daily                     | 4.27     | 2.46   | 2.49    |
| Weekly                    | 5.43     | 2.62   | 2.79    |
| Weekly and daily          | 4.07     | 2.37   | 2.42    |
| Weekly, daily and hourly  | 5.83     | 2.34   | 2.35    |

Table II. Boosters NRMSE values in percentages in 24 hour forecast
The comparison is not completely fair due to following reasons:

- **ResNet** does not in fact solve the same problem we have stated. The model is suitable for a little bit easier problem where one forecasts energy consumption of each of the hours on the next day given all the energy consumptions of the current day. It utilizes therefore 0-23 more recent observations that we do not utilise depending on the forecast hour.

- Architecture of **SBN** is tuned for this particular data set while **ResNet** is used in plug and play style. However, our network architecture is not very sensitive to changes of each submodel architecture.

The target metrics for comparison is shown in Table IV. **ResNet** solutions seems to perform quite well for plug-and-play solution but when there are very much training data available. However, when reducing the amount of training data it starts to overfit and performance decreases dramatically. However, our **SBN** solution behaves very well on even a relative short training data size.

### IV. Conclusion and the Future Work

We proposed a **SBN** architecture for short term energy load forecasting. The architecture is general in a sense that it could be used for other similar time series forecasting domains also. We continue studying it in short term energy load forecasting. So far, we evaluated the architecture with only one simple data set. Next, we apply it to more complex data sets that contains changing phenomena of different time scales such that one would need really to stack multiple boosters as it is described in the architecture of **SBN**. However, as shown in this paper, stacking the multiple boosters did not cause harm for the forecast even if the time series did not contain very much such complex phenomena.

This paper covered different training options of the **SBN** architecture. However, one interesting future issue was left without mention. As is described in the paper, **SBN** is composed of **Instant forecaster** and different boosters. Boosters solves very general univariate time series forecasting problem while **Instant forecaster** makes the dataset specific prediction. Therefore it could be possible to train or pre-train the boosters using totally different energy data sets and train only **Instant forecaster** with the target data set. This way we could utilize transfer learning paradigm and make the energy forecasts with even reduced amount of training data from target building.

### REFERENCES

1. U. Berardi, “A cross-country comparison of the building energy consumptions and their trends,” *Resources Conservation and Recycling*, vol. 123, pp. 230–241, 07 2017.
2. K. Amasyali and N. M. El-Gohary, “A review of data-driven building energy consumption prediction studies,” *Renewable and Sustainable Energy Reviews*, vol. 81, no. P1, pp. 1192–1205, 2018.
3. A. Bagnasco, F. Fresi, M. Saviozzi, F. Silversto, and A. Vinci, “Electrical consumption forecasting in hospital facilities: An application case,” *Energy and Buildings*, vol. 103, pp. 261 – 270, 2015.
4. Y. T. Chae, R. Horesh, Y. Hwang, and Y. M. Lee, “Artificial neural network model for forecasting sub-hourly electricity usage in commercial buildings,” *Energy and Buildings*, vol. 111, pp. 184 – 194, 2016.
5. S. Wong, K. Wan, and T. Lam, “Artificial neural networks for energy analysis of office buildings with daylighting,” *Applied Energy*, vol. 87, no. 2, pp. 551–557, 2010.
6. G. Escriv-Escriv, C. Ivarez Bel, C. Roldn-Blay, and M. Alczar-Ortega, “New artificial neural network prediction method for electrical consumption forecasting based on building end-uses,” *Energy and Buildings*, vol. 43, no. 11, pp. 3112–3119, 2011.
7. Y. Zhang and Q. Li, “A regressive convolution neural network and support vector regression model for electricity consumption forecasting,” *CoRR*, vol. abs/1810.08878, 2018.
8. R. K. Jain, K. M. Smith, P. J. Culligan, and J. E. Taylor, “Forecasting energy consumption of multi-family residential buildings using support vector regression: Investigating the impact of temporal and spatial monitoring granularity on performance accuracy,” *Applied Energy*, vol. 123, no. Complete, pp. 168–178, 2014.
9. R. K. Jain, T. Damoulas, and C. E. Kontokosta, *Towards Data-Driven Energy Consumption Forecasting of Multi-Family Residential Buildings: Feature Selection via the Lasso*, pp. 1675–1682.
10. D. L. Marino, K. Amarasinghe, and M. Manic, “Building energy load forecasting using deep neural networks,” *CoRR*, vol. abs/1610.09460, 2016.
11. W. Kong, Z. Dong, D. Hill, F. Luo, and Y. Xu, “Short-term residential load forecasting based on resident behaviour learning,” *IEEE Transactions on Power Systems*, vol. PP, pp. 1–1, 03 2017.
12. W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, “Short-term residential load forecasting based on lstm recurrent neural network,” *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan 2019.
13. K. Amarasinghe, D. Marino, and M. Manic, “Deep neural networks for energy load forecasting,” *IEEE Transactions on Power Systems*, vol. PP, pp. 1–1, 03 2017.
14. K. Yan, X. Wang, Y. Du, N. Jin, H. Huang, and H. Zhou, “Multi-step short-term power consumption forecasting with a hybrid deep learning strategy,” *Energies*, vol. 11, pp. 3089, 11 2018.
15. E. Mocanu, H. Nguyen, M. Gibescu, and W. Kling, “Deep learning for estimating building energy consumption,” *Sustainable Energy, Grids and Networks*, vol. 6, pp. 9199, 6 2016.
16. K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, and J. He, “Short-term load forecasting with deep residual networks,” *IEEE Transactions on Smart Grid*, pp. 1–1, 2018.
17. K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *CoRR*, vol. abs/1512.03385, 2015.

---

1. [https://github.com/yalickj/load-forecasting-resnet](https://github.com/yalickj/load-forecasting-resnet)