House model machine learning

Marijn JOngerden
HAN University of Applied Sciences
Arnhem, The Netherlands

January 11, 2022

1 Introduction

The developed house model (put a ref here) allows to simulate the heating process of a given house. Based on weather information such as the outside temperature and solar irradiation, house specifications, and other input factors it is possible to compute the details on the temperature evolution.

The model needs many details to run. These details will be hard to get a hold of in practice.

It would be interesting to see whether it is possible to use machine learning to abstract from the details of that are used to create the model and use only limited amount of input data to create a model that can predict the heat demand of a house.

2 Background

3 Machine learning approach

Intro to the section...

3.1 LSTM-model

We create a Long short-term memory (LSTM) model for predicting the heat demand. LSTM is a type of recurrent neural network (RNN). These models are well suited for time series data, as we have here. In a RNN the output of a layer in the network can be fed back to be used as input of an earlier layer in the network. In this way, dependencies on the history can be taken into account in the model. LSTM is a specific type of RNN, designed such that the network learns which historical data is needed to ¨rememberüsing the memory cell, and which data can be ¨forgottenüsing a so-termed forget-gate. Details on the working of the LSTM can be found in ...

something on the sequence length and batches ...

3.2 input data selection

In this first exploration of the potential of using a machine learning model to predict the heat demand of a household we make use of data generated with the house model. The house model provides a heat demand profile for a full year. The heat demand is based on weather data, temperature and solar irradiation, construction parameters of the house, and some behavioral parameters, such as the number of people present in the house and the thermostat setting. The model is described in detail in **??**. The house is modeled as a network of heat capacitors, between which energy can be exchanged. The most simple house model uses 2 heat capacitors, one for the air in the house and one for the walls. Heat can be exchanged between the two capacitors, and the outside air. Essentially this model approximates the house as a single compartment with its walls. A more detailed representation of the house can be made by using multiple heat capacitors, for example one for each floor or room. Several versions of the specific model have been developed. Initially we make use of the data generated by the Matlab implementation, which has a time granularity of one hour. Later we shift to the more detailed Python implementation of the model, which has more flexibility and a higher time granularity.

The input features we want to use for our model should also be easily available in practice. The input features we select to create our model are:

- Outside temperature (T_{out}); we can assume this parameter is easily available through either a local thermometer or an online service providing data of a nearby weather station. Alternatively weather forecast date might be used.
- temperature of the house (T_{house}); this should be available through the thermostat. In the house model this is the temperature of air. In case of a model that has multiple compartments, we use the air temperature of the main compartment.
- thermostat set point (SP), this should be available though the thermostat. Potentially, these values are even available for future values, since often programs are created to set the thermostat values.
- solar irradiation (Q_{solar}), although these values may not be straightforward to obtain for a specific location, an estimate value may be obtained from a local weather station.
- heat demand (Q_{demand}), the historic values of the heat demand will be used to predict the future values.

More features could be available from the model, such as the heat generated by the appliances and inhabitants or the temperature of the walls. However, we hope to obtain a machine learned model that can abstract from these details and still give a good prediction for the heat demand for the coming time period.

NOTE: It might be interesting to also look at the capabilities to predict the house temperature as well. For the other inputs a future value may be obtained from either the weather forecast or the thermostat program. Taking the temperature predictions into account might be helpful for a model predictive control approach?

3.3 preprocessing steps

The input data is loaded from an excel sheet that contains all the features. When we want to create a model the data needs to be spit in a training and a test set. The first 70% of the data will be used for training the mode and the last 30% will be used for testing.

After splitting the data, the training set is normalized, by subtracting the mean of all feature values (μ) and dividing by the standard deviation (σ) (using the standardscaler),$x_{norm} = \frac{x-\mu}{\sigma}$. The output is normalized as well.

After the normalization of the data, the data needs to be prepared for the LSTM. For each output value, heat demand at time t, a sequence of historical data of input values is needed, time steps $t - N \dots t - 1$. (*NOTE: the value of N, the number of historical time steps needs to be determined. For the hourly data* $N = 12$ *seems to give good results. A logical choice would be to set N such that 24 hours of historical data is used to predict the next time step, since a high correlation with 24 hours earlier is to be expected.*)

4 Python implementation

5 results

6 discussion