Price Predictions on Airbnb Accomodations in Oslo, Norway

Marei Freitag, Joel Beck

Georg-August-University of Göttingen

21.02.2022

Table of Contents

Introduction

Aims of this work:

- Establish a deep learning approach to predict the price of an Airbnb accomodation per night in Oslo, Norway
- Focus on explainability and interpretability

$\rightarrow$ Underlying data: provided by Airbnb, contains various information about the listings in Oslo

Table of Contents

Feature Engineering: Images

- Use transfer learning on a pretrained CNN (`ResNet18`) with the first 5 images per listing as input data
- Added Fully Connected Network at the end containing three layers and `ReLU` activation functions to be sure the CNN is able to generalize
- Also implemented CNN manually as a benchmark model to compare the results

Results:

- pretrained `ResNet18` achieved a Mean Absolute Error of 579 NOK (approx. 58 Euros) on the validation set
- But correlation of the CNN predictions with the true price is 0.41

Image Predictions

Figure: CNN example predictions

Feature Engineering: Reviews

- Language: Detect language of each review
- Sentiment analysis: Get the sentiment of each review

New features per listing:

1. Number of reviews
2. Median review length
3. Number of different languages of the reviews as well as a list of the different languages
4. Fraction of Norwegian and English reviews
5. Ratio of negative reviews to the total number of reviews

Wordclouds of the Reviews

Figure: Wordclouds in English and Norwegian

Feature Selection & Data Cleaning

Feature Selection:

1. Manually selected features based on background knowledge, correlation analysis and the number of missing values
2. Adjusted these features by analyzing the results of different feature selection algorithms and fitted auxiliary linear regression

Data Cleaning:

- Converting data types
- Splitting text-based variables into more convenient numeric or boolean features
- Aggregating rare categories of categorical variables into one larger *Other* group to stabilize estimation
- One-Hot encoding of categorial variables and standardization of numerical variables

Table of Contents

Classical Models

1. **Linear Regression**: simple, well understood in terms of underlying theory and highly interpretable.
2. **Ridge Regression**: still very interpretable with a closed form analytical solution; one hyperparameter
3. **Random Forest**: very flexible model with many hyperparameters determining e.g. the number of regression trees and the tree depth, but can be applied to many contexts and often works 'out of the box'
4. **Histogram-Based Gradient Boosting**: modern and fast tree-based gradient boosting algorithm; large number of tunable hyperparameters

Neural Network: Model Architecture

- Linear input layer (about 60 features)
- 6 intermediary **blocks** with 64, 128, 256, 128, 64 and 8 output features:
  - Residual connection
  - Linear layer with BatchNorm, `ReLU` activation function and dropout
- 1 output neuron

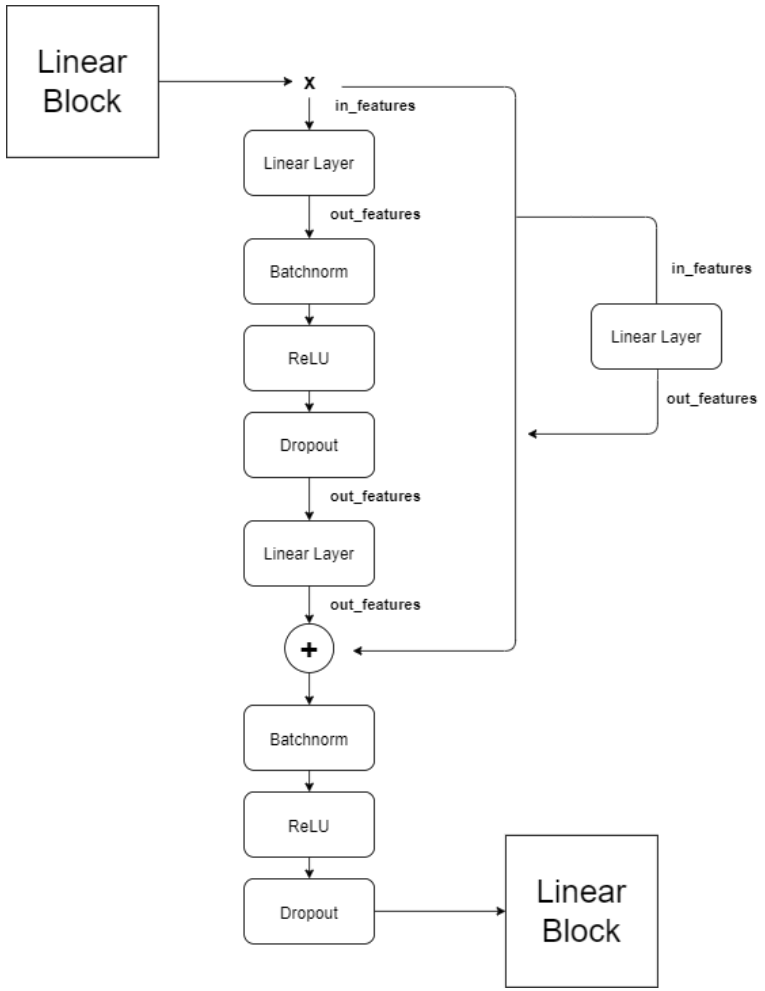

Figure: Linear Block in the FC-NN

Neural Network: Model Training

- Optimizer: `Adam` with learning rate set to 0.01
- Loss function: *Mean Squared Error* Loss
- Epochs: Number of epochs vary; stopped training if Loss stagnated or model began to overfit

Most impactful hyperparameter: **Dropout rate**

- high influence of the network's generalizatioin availability
- model overfitted significantly by setting dropout rate to zero $\rightarrow$ that shows the current model structure is flexible enough to model the task properly
- increasing the rate leads to higher training MAE but also improves the model's performance on the validation set

Table of Contents

Test - Slide 1

Table of Contents

Test - Slide 2

Thanks for listening!

Questions?