Seminar Paper
Energy Economics

Alexander Peytz

Spring 2024

Contents

1 Introduction

2 Methodology

2.1 Deep Neural Network

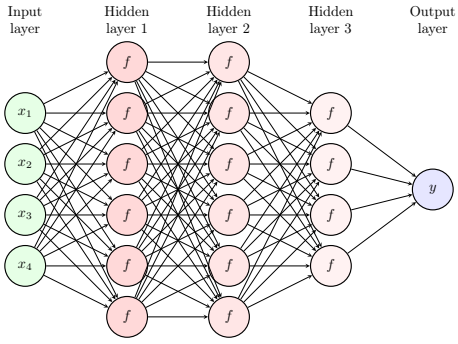
2.1.1 Overview

Artificial neural networks (ANN) are considered and analyzed as a forecasting method. Neural networks consist of a vast class of models, applicable to a number of tasks which essentially boil down to whether the input and output can be represented numerically, these include image and audio classification, language processing, and much more. If the problem can be stated as an optimization problem, then a neural network method is applicable, even for complex formulations. See **abiodun2018state** for a comprehensive review on the applications.

In a time-series setting the neural network is applicable given their adaptive nature, allowing for a multitude of effects between the predictors and the target variable. Multiple types of ANN's are possible candidates for electricity price forecasting, and have been implemented in the literature with differing success. These include Feed forward neural networks (FNN), recurrent neural networks (RNN), such as LSTM and GRU, Convolutional neural networks (CNN) and many more.

This paper will analyse the use of a deep feed forward neural network since this architecture is able to handle a variety of complex patterns in multivariate time-series data and remains computationally feasible, see **DLforTimeSeries2020**. The transition from a FNN to deep neural network (DNN), leads to some intertemporal advantages by including multiple sequential hidden layers, while the use of the feed forward structure leads to a solution which more closely resembles a functional mapping of explanatory input variables to a target output. This structure consists of a set of input variables, which are the predictors, which are passed along with a set of weights and a constant to a number of 'hidden' layers, consisting of neurons which apply an activation function f, and passing the output to the next layer. Finally the output of each neuron from the last layer in the sequence are aggregated into the output neurons see figure **??**.

Figure 1: Structure of a feed forward neural network



We define a deep neural network as a network consisting of multiple hidden layers, and a wide network consists of a large amount of neurons. The total amount of trainable parameters of a deep neural network is given by:

$$\sum_{i=0}^{L-1} (n_i + 1) \cdot n_{i+1} \tag{1}$$

Where L denotes the amount of layers including the input and output layers, and n denotes the amount of neurons in the layer.

2.1.2 Activation function

The activation function is essential to the network. It introduces non-linearity in the output of a neuron. This non-linearity allows the network to handle non-linear patterns and interactions between features, and between layers, enabling the model to learn and perform tasks that go beyond simple linear relationships. Most of the literature on electricity price forecasting implements networks using rectified linear unit (ReLU) as activation function which is a popular choice of activation function of the following form:

$$\max(0,x) \tag{2}$$

ReLU has however been found to suffer from a set of disadvatanteges. The neurons can suffer from a problem known as "dying ReLU." This issue occurs when neurons become inactive and only output zero due to negative input values, potentially inhibiting the estimation of the gradient during optimization. See **lu2019dying** for more on this. We propose the use of the GELU (Gaussian Error Linear Unit) activation function This activation function mitigates the issue by providing a smooth continuous curve that does not zero out negative values, allowing for a continuous gradient for inputs that would otherwise deactivate ReLU neurons, see **hendrycks2023gaussian** for more on this.
GELU has the functional form:

$$x \cdot \frac{1}{2}\left(1 + \frac{2}{\pi}\int_0^{\frac{x}{\sqrt{2}}} e^{-t^2} dt\right) \tag{3}$$

We likewise test the use of other 'modern' activation functions such as the use of the SiLu and the Mish activation function. See **misra2020mish** and **ramachandran2017searching** for more on these actiation functions.

2.1.3 Regularization and dropout

Regularization and dropout are techniques used to prevent overfitting, which could lead to worse out of sample prediction. Regularization imposes penalties on the complexity of the model, typically by adding a term to the loss function that discourages large weights. Dropout randomly deactivates a proportion of neurons during training, which encourages the network to develop more robust features that are not reliant on any single neuron see **Srivastava2014**. We consider using L1 and L2 regularization penalties. L1 regularization applies a penalty based on the absolute value of the coefficients, and L2 regularization applies a penalty based on the square of the coefficients. L1 which is commonly known from Lasso regression, applies feature selection by setting irrelevant and insignificant coefficients to 0. This type of regularization is appropriate given the amount of feature engineering done. L2 regularization is most commonly known from Ridge regression and introduces a penalty which shrinks the magnitude of coefficients without setting them to 0.

2.1.4 Data normalization

Input features are scaled using the RobustScaler from the Scikit library available in Python. Without scaling, features with larger magnitudes could disproportionately influence the model, leading to suboptimal training outcomes and difficulties in convergence. By normalizing features to a similar scale, we enable the model to train more efficiently and often achieve better performance. This scaler scales each feature based on the median and the interquartile range, which is effective for datasets that might contain outliers. Since the electricity price might suddenly drop or increase, this scaler is appropriate as these will not drastically influence the core data.

2.2 Feature selection and transformations

2.2.1 Overview

Feature transformations, known in the neural network literature as feature engineering, defines the transformations done to the predictors (features), of the data structure in order to better predict the target variable. This ensures that the transformations are available to the neural network, and that the network picks up on any patterns seen in the new variables. This can therefore be seen as 'a gentle push' of the network in the 'direction' of the transformations. Another way to look at this is to ensure that the network is able to pick up on specific structures of the predictors. Furthermore, complex layer types, which could allow the network a better temporal understanding of the data, such as LSTM's, aren't feasible on the hardware available to the author. Forcing feature transformations, such as lags, enforces a temporal structure on the forecast, allowing for some of the benefits of RNN's which otherwise would be more applicable to time series data. Likewise applying rolling statistics, such as the variance, ensures that these are also present in the network. The following is a brief summary of the transformations considered.

2.2.2 Simple feature transformations

We consider lagged features, which pick up any auto regressive explanatory power in the dataset.
We also consider adding differences between variables, either between lags or between predictors, in order to ensure that the network can pick up on time variance and marginals.
We likewise consider rolling summary statistics, such as a rolling mean and rolling variance transformation, within a given window. This is a smoothed representation of the summary statistics which should capture generalised patterns in the price movements.
Interaction terms are also considered. Adding interaction terms as an explicit input variable ensures that the combined effect of the input variables is appropriately taken into account by the model. These terms can reveal relationships that are not apparent when considering the features independently.
Autocorrelation features measure the correlation of the time series with its own past values, identifying patterns such as seasonality or cyclic behavior.

2.2.3 Wavelet transform

The wavelet transform is a transformation akin to the Fourier or Gabor transformations, in which signals, such as a time-series, can be represented as the sum of component signals. The wavelet transform decomposes the time series into components that capture information at different scales such as high-frequency details and low-frequency trends. Using the DB1 wavelet allows for a simple breakdown, capturing abrupt changes in the signal. This transformation therefore breaks the tiem-series into components which make less sense intuitively, but can capture different patterns in the dataset. Wavelets are as of yet not as implemented in financial analysis as Fourier analysis but is offers a time analysis as well as a frequency analysis. for more see **zavanelli2023wavelet**.

2.3 Performance metrics

In order to quantify the performance of the neural network, two different metrics are considered.

2.3.1 Mean absolute error

In order to evaluate the performance of the forecasts we introduce the mean absolute error (MAE), which measures the mean forecast error between the actual data and the predictions.

$$\text{MAE} = \frac{\sum_{i=1}^{n} \mid y_i - \hat{y}_i \mid}{n} \tag{4}$$

Where y_i are the actual values and $\hat{y}_i$ denotes the predicted values.
This is a simple and effective metric of evaluating the effectiveness of the network.

2.3.2 Mean absolute percentage error

The mean absolute percentage error (MAPE) measures the forecast accuracy relative to the actual values.

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n} \mid \frac{y_i - \hat{y}_i}{y_i} \mid \tag{5}$$

Where y_i are the actual values and $\hat{y}_i$ denotes the predicted values.

2.4 A description of the day ahead market

The danish energy system is split into two areas, DK1 and DK2, respectively west and eastern Denmark. The danish energy market can be further split into two markets, the wholesale market and the retail market.
Since this analysis is focused on the day ahead prices, this section will not cover the retail market, since these prices are formed on the wholesale market.
The Danish wholesale market is comprised of electricity producers and electricity distributers. These actors trade on a electricity exchange comprised of multiple european countries.

2.5 Price formation

The day-ahead market, also known as the spot market, is where electricity producers and whole sale consumers, e.g traders and distributers buy and sell electricity for delivery at hourly intervals the following day. This occurs on electricity exchanges, and is structured as a blind auction. Electricity suppliers and buyers submit buy and sell order bids for the next day's electricity supply, this can occur until 12:00 (CET) when the market closes.
Following the the market close, the market clearing price is calculated, matching buy and sell bids, such that buyers pay up to their disclosed willingness to pay, and sellers sell at prices higher than or at their disclosed willingness to sell. This occurs at 13:00 (CET). This is the Day-Ahead price. For a more comprehensive review of the Danish electricity market see **Energinet2019**.
From the close of the day ahead market, trade of the electricity can still occur on the intraday market, from 15:00 the day prior and up until 45 minutes prior to the hour of delivery, however we will not go further into this, given the scope of this paper.

3 Data

Data stems from Entsoe transparancy database and DMI Climate Data API. This covers 01.01.2018 - 01.04.2024 for the two bidding areas of Denmark, DK1 and DK2 as well as the closest bidding areas of neighbouring countries, Norway, Germany and Sweden. Data has been pre-processed, removing missing observations which stem from summer solstice, as well as removing missing observations in the beginning of the observed period.
Climate data covers a single municipality in each of the two bidding areas. For DK1, the chosen municipality was Copenhagen, whereas for DK2 Aarhus was chosen.
Hence a complete dataset has been established for the period 01.01.2018 - 01.04.2024. After pre-processing the data the proposed feature engineering is implemented, adding interaction terms across the entire parameter space. Any other proposed feature is likewise implemented on the day ahead price. This results in 743 predictive parameters which can then be feeded into the network. Lags implemented are in the interval of $D-1$, $D-2$, $D-3$, $D-7$ and $D-30$ with D corresponding to days.

Table 1: Available variables in the dataset

Area	Variable	Source
DK1	Day_Ahead_Price	Entsoe
DK1	Forecasted Total_Generation	Entsoe
DK1	Forecasted Total_Load	Entsoe
DK1	Forecasted Solar_Generation	Entsoe
DK1	Forecasted Wind_Generation	Entsoe
Copenhagen	Forecasted mean_temp	DMI
Copenhagen	Forecasted mean_wind_speed	DMI
DK2	Day_Ahead_Price	Entsoe
SE4	Day_Ahead_Price	Entsoe
NO2	Day_Ahead_Price	Entsoe
DE	Day_Ahead_Price	Entsoe
DE-LU and DK1	Forecasted transfer capacities	Entsoe
Global	Spot Coal Price (USD)	Business Insider
Global	Spot Natural Gas Price (USD)	Business Insider
Global	Spot Crude Oil Price (USD) (Brent)	Business Insider
Global	Spot Heating Oil Price (USD)	Business Insider

Hence the proposed neural network, predicts the day ahead price, based on price movements of neighbouring areas, supply and demand movements, prior price movements and climate data.
This is a more comprehensive dataset than what is generally found in the litterature on price forecasting.

3.1 Predictors

The predictors displayed in **??**, can be grouped into the following classes of predictors, price data, load data, generation data, climate data and transmission data.
Below is a brief explanation for the incorporation of each data type:
Price data: This includes day-ahead electricity prices for DK1 and different market areas. Prices inherently reflect the market dynamics and interdependencies between neighboring electricity markets. Prices in these areas can influence DK1 prices due to cross-border trading and shared supply-demand shocks.
Generation and Load Data: Generation data reflects the supply side of the market, which directly influences the price as noted earlier. Generation data is split in its renewable components, due to their large share of the total danish electricity production, see **IEADenmark2023** for a comprehensive review of the danish energy sector. Load data reflects the demand side of the market and is therefore likewise incorporated as a predictor of the price movements.
Climate data: Mean temperature and wind speed data are included as they significantly affect both demand and renewable energy production. Temperature impacts the demand for heating or cooling, and Wind influences wind power generation. Since renewable energy sources like wind and solar are weather-dependent, their unpredictability due to climate conditions can cause significant volatility in electricity prices, which is then directly incorporated into the model.
Transmission data: Forecasted transfer capacities reflects the physical limitations and opportunities for electricity trade across bidding areas. Constraints in transmission capacities can lead to regional price differences.

3.2 Timeliness

We now consider the publication time of the data available for forecasting. Since the day ahead price for DK1 and DK2 is published at 11:00 the day prior, any data which is publically available after this point in time, cannot be used for forecasting purposes. We therefore decide to only use data available 2 days prior to the publication of the day ahead prices. The resulting network will therefore be able to forecast the prices published on a wednesday, using data available from monday and so on.

3.3 Descriptives

4 Results

4.1 Network Architecture

4.2 Application to the DK1 market

4.3 Performance across time-frames

4.4 Large network vs Specifik network

4.5 Rolling recalibration

4.5.1 Large network results

4.5.2 Small network results

5 Discussion

5.1 Applicability and feasibility

6 Conclusion

The following references are tentative*