IM1102-232433M - Deep Neural Engingeering assignment 2

Modifying the attention mechanism of transformers for time series forecasting

Arne Lescrauwaet - Joachim Verschelde - Alexander Van Hecke

April 7, 2024

Introduction

This report details the steps taken by Arne Lescrauwaet (852617312), Joachim Verschelde (852594432) and Alexander Van Hecke (852631385) for the second assignment of the 2023 Deep Neural Engineering course organised by the Open University (1).

For this assignment we look at different attention mechanisms in transformers (2) for use with time series data. The attention mechanism enables a transformer model to selectively focus on relevant parts of the input data. The goal is to be able to capture long range dependencies and relationships between items of the input data. This is particularly important for time series data containing recurring patterns, e.g. hourly traffic counts on busy highways and hourly power consumption of nations. We expect these types of data to contain clear and recurring patterns (i.e. traffic will typically be lower during weekends) and we want an attention mechanism to capture these patterns. In addition to capturing recurring patterns, we would also like to be able to capture the "local context" of a pattern to predict new values. That is, when encountering an event that is similar to a past event, we want to take the outcome of that past event into account in our prediction.

Different kinds of attention mechanisms exist. Convolutional self-attention is introduced in (3), which aims to capture the local context of input events, but does this using a symmetric convolution, thereby taking both input data leading to a particular event and the outcome of that event into account. A dual-stage attention mechanism is used for a Recurrent Neural Network (RNN) architecture in (4), using an input attention mechanism in an encoder step, and a temporal attention mechanism in a decoder step.

Even though transformers were originally designed in the field of natural language processing (NLP), a lot of work has been done to use transformers with time series data. An overview of different ways to adapt transformers to time series data is given in (5). The time2vec encoding mechanism is introduced in (6). The authors of this paper use transformer models to predict stock prices, and claim these models can be used both for short and long term predictions. The effectiveness of applying transformers to time series data is tested in (7).

The original transformer architecture introduces a quadratic time and space complexity. Much work has been done to improve on this. The LogSparse transformer is introduced in (3), which reduces the memory cost to $O(L(\log L)^2)$. The informer model (8) even achieves $O(L\log L)$ memory complexity. In this report we will focus on attention mechanisms in the context of time series forecasting, ignoring space and time complexity of the transformer algorithm.

Goal

In this paper, we focus on using transformers for time series forecasting. We aim to compare different attention mechanism and determine which mechanism best captures the outcome of past events. We formulate a first research question :

> **RQ 1 : When comparing regular self-attention, convoluted self-attention, asymmetric convoluted self-attention and eigenvector based self-attention, which mechanism best predicts future values using mean square error (MSE) as metric?**

The Elia dataset used is fully described in the dataset description section. It not only contains time series data, but also day+1 and day+7 predictions of the same data. We formulate a second research question :

RQ 2 : Is the MSE of a transformer model better than the Elia prediction model?

Firstly, this report will look at the characteristics of the dataset used and discuss pre-processing steps. Then, we will consider several attention mechanisms, discuss design and implementation details and finally evaluate the performance of these attention mechanisms on the dataset.

Data analysis

Dataset description

We use data from Elia (9), which operates the electricity transmission network in Belgium. In particular, we use the solar power forecast datasets. These contain time series of actual measured power in megawatt (MW), and also day+1 and day+7 predictions of solar power output in MW. Data is available for a period of 12 years (February 2012 until now) in monthly datasets. Measurements and predictions are recorded every quarter of an hour. The measured value is always the amount of power equivalent to the running average measured for that particular quarter-hour. The layout of the dataset is fully described here (10). We recap the most important points in Table 1.

Table 1: Features captured per quarter-hour in (10)

feature	description	range
DateTime	Date and time per quarter hour	[00:00 - 24:00] in quarter hours
Measurement	Measured solar power production in MW	[0.0 - 6000.0]
Day+1 prediction	D+1 solar power forecast in MW	[0.0 - 6000.0]
Day+7 prediction	D+7 solar power forecast in MW	[0.0 - 6000.0]

Data general properties

Data is not normally distributed but highly regular and contains obvious day - night recurring patterns. Since we are using solar power production data, data typically shows no values in the early morning, building towards a peak around noon, and then slowly reducing values towards the evening. This is illustrated in Figure 1.

There are obvious differences in solar power generation between summer months and winter months, but the general pattern remains the same, as illustrated in Figure 2.

Data pre-processing

The Elia data (10) is very fine grained and contains $24 * 4 = 96$ measurements per day, resulting in $30 * 24 * 4 = 2880$ measurements for a 30 day month. In order to be able to limit memory and computational resources, we have added the possibility to aggregate these dataset. Possible choices are **(i)** no aggregation, **(ii)** hourly aggregation, **(iii)** aggregation every 4 hours (starting from 00:00, resulting in 6 values per day), and finally **(iv)** aggregation per day. Aggregation is done by averaging the values in the selected timeframe.

Elia provides a lot of historical data, going back more than 10 years in the past. We selected 10 years of data (2014-2023), only selecting years containing data for all months. Furthermore, we wanted to investigate scenarios making sense for the data used. This means we did not want to mix data of summer months (very high solar power production) with data of winter months (very low solar power production). We added a selection mechanim for **(i)** taking data of one particular month across all 10 years, and **(ii)** taking data of one particular season (winter, summer) of a single year. When selecting a single month across all years, all values were concatenated into a single dataseries. When selecting a season, e.g. summer, all values of the different months of the season were concatenated into a single dataseries.

Input length L has to be chosen carefully in basic transformer architectures because of the quadratic complexity in L. Taking too few days into acount, it will be difficult to spot similar events in the past. Taking too many days into account, it will be prohibitely expensive in terms of memory and computational resources to train and evaluate the model. The dataset and dataloader implemented allowed for a selection of 5, 10 or 20 days.

This is summarized in Table 2.

Table 2: Possible pre-processing steps

step	description	options
aggregation	Reduce number of values by averaging	no aggregation, hourly, 4-hourly, daily
selection	Selection of specific months	same month across 10 years, season in one year
padding	Selection of input length in days	5, 10, 20 days

TODO nachtelijke uren eruit halen?

Outlier analysis

A visual outlier analysis yielded no abnormal or obiously wrong values. This makes sense, as the data contains actually measured solar power. Therefore, no values were discarded.

Methodology and Implementation

Research methodology

We started by examining the dataset provided (10). Outlier analysis yielded no results, and we performed a number of standard checks on the quality of the data and decided not to exclude any data from the dataset.

Given a basic transformer architecture, we implemented a number of attention mechanisms to investigate influence on prediction MSE. Models were tuned using appropriate hyperparameters using TODO TODO TODO cross-validation. Several datasets were generated, properly aggregating data and using both seasonal and monthly historical scenario's. Each model was then used to to predictions on these data sets. MSE was used as the loss metric.

TODO beschrijven hoe split training / validation / test set.

Design elaboration

We decided to evaluate the following attention mechanisms (Table 3) :

- regular self-attention (AM-1). This is the mechanism described in the original transformer paper (2).
- convoluted self-attention as described in (3) (AM-2). This is the mechanism described in (3). It generalizes the regular self-attention mechanism and uses a 1D convolution to transform the Query (Q) and Key (K) values before using them in the transformer architecture.
- asymmetric convoluted self-attention (AM-3). This is a variation of the mechanism described in (3). Whereas (3) uses a symmetric convolution, here we use a (right-)asymmetric convolution to transfer Q and K values before using them in the transformer architecture.
- eigenvector based self-attention (AM-4). This uses eigenvectors as a measure of similarity between keys and values to determine where to direct attention.

Table 3: Attention mechanisms

attention mechanism	abbreviation
regular self-attention	AM-1
convoluted self-attention	AM-2
asymmetric convoluted self-attention	AM-3
eigenvector self-attention	AM-4

TODO extra features / embedding

-> feature + positional encoding + one-hot encoding van dag of maand of week “temporal encoding”

TODO verder verduidelijken

We start by splitting the dataset (feature X and target vector y) in a training set (35%, X_{train} and y_{train}), a validation set (35%, $X_{\text{validation}}$ and $y_{\text{validation}}$) and a test set (30%, X_{test} and y_{test}). Then, a grid search with cross-validation is used to train the model on the training data and validate it against the validation set. The grid search is parameterized by relevant parameters specific to the self-attention mechanism, see Table 4. Attention heads are varied between 2 and 8 in increments of 2. For the convolution based attention mechanisms, we investigate kernel sizes ranging from 3 to 9 in increments of 2. We use MSE as a measure to optimize for.

Table 4: hyperparameters used for the transformer models

attention mechanism	hyperparameter	range
AM-1	attention head	range(2, 8) step 2
AM-2	attention head	range(2, 8) step 2
AM-2	kernel size	range(3, 9) step 2
AM-3	attention head	range(2, 8) step 2
AM-3	kernel size	range(3, 9) step 2
AM-4	attention head	range(2, 8) step 2

In each iteration, we run the grid search using the training data, and predict against the test data ($y_{\text{predicted\_test}}$). All predictions are stored for later analysis.

This entire design is repeated for a number of different scenarios. We detail these in Table 5.

- scenario summer-season : In this scenario, we aim to investigate recurring events in the most sunny season of one year. We concatenate all data of June, July, and August of 2023.
- scenario winter-season : In this scenario, we aim to investigate recurring events in the least sunny season of one year. We concatenate all data of December, January and February of 2023.
- scenario summer-month : In this scenario, we aim to investigate recurring events in the most sunny month of all years (period 2014-2023). We concatenate all data of August for years 2014-2023.
- scenario winter-month : In this scenario, we aim to investigate recurring events in the least sunny month of all years (period 2014-2023). We concatenate all data of February for years 2014-2023.

Table 5: learning scenarios

scenario	description
summer-season	concatenation of June, July, and August of 2023
winter-season	concatenation of December, January, and February of 2023
summer-month	concatenation of August data of period 2014-2023
winter-month	concatenation of February data of period 2014-2023

Implementation

All code and data is available in a github repository (11). All deep learning models were implemented using the pytorch python package.

Evaluation and Results

Evaluation

TODO beschrijven wat we exact willen meten en hoe dit te meten (loss) (accuracy?)

To evaluate whether ... TODO ... self-attention ... , we formulate the following H_0 hypothesis :

> **H_0 : A self-attention mechanism using XYZ is not better at predicting ... than regular self-attention .**

If the p-value is below $\alpha = 0.05$, we can reject H_0 and accept the alternative hypothesis, that there is indeed a difference between the TODO.

-> vergelijken met base line voorspellingen elia? -> regressieanalyse van de residuals.

Results

Scenario 1 : summer-season

This scenario uses data for June, July and August of 2023 for forecasting. Results are summarized in Table 6.

Table 6: one sample t-test to determine whether TODO

mechanism	metric mean	AM-1 mean	t-test value	p-value	H_0 rejected
MA-2	0.9580	0.9864	-1.77E+01	2.29E-32	yes
MA-3	0.9858	0.9864	-1.84E+01	1.17E-33	yes
MA-4	0.9782	0.9864	-2.58E+01	9.62E-46	yes

TODO resultaten beschrijven

Scenario 2 : winter-season

TODO idem hierboven

Scenario 3 : summer-month

TODO idem hierboven

Scenario 4 : winter-month

TODO idem hierboven

Conclusions and Discussion

In this study, we have used xyz dataset and pre-processed thus and thus.

We evaluated x self-attention mechanisms, x, y and z in x different scenarios. Results were :

- result 1
- result 2

TODO some discussion

TODO future work

References

1. Startpagina - IM1102-232433M - Deep Neural Engineering [Internet]. [cited 2024 Mar 21]. Available from: https://brightspace.ou.nl/d2l/home/8636
2. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is All you Need. In: Advances in Neural Information Processing Systems [Internet]. Curran Associates, Inc.; 2017 [cited 2024 Mar 6]. Available from: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html
3. Li S, Jin X, Xuan Y, Zhou X, Chen W, Wang YX, et al. Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting. In: Advances in Neural Information Processing Systems [Internet]. Curran Associates, Inc.; 2019 [cited 2024 Mar 10]. Available from: https://proceedings.neurips.cc/paper/2019/hash/6775a0635c302542da2c32aa19d86be0-Abstract.html

4. Qin Y, Song D, Chen H, Cheng W, Jiang G, Cottrell GW. A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence [Internet]. Melbourne, Australia: International Joint Conferences on Artificial Intelligence Organization; 2017 [cited 2024 Mar 6]. p. 2627–33. Available from: https://www.ijcai.org/proceedings/2017/366

5. Wen Q, Zhou T, Zhang C, Chen W, Ma Z, Yan J, et al. Transformers in Time Series: A Survey [Internet]. arXiv; 2023 [cited 2024 Mar 7]. Available from: http://arxiv.org/abs/2202.07125

6. Muhammad T, Aftab AB, Ibrahim M, Ahsan MdM, Muhu MM, Khan SI, et al. Transformer-Based Deep Learning Model for Stock Price Prediction: A Case Study on Bangladesh Stock Market. International Journal of Computational Intelligence and Applications [Internet]. 2023 Sep [cited 2024 Mar 16];22(03):2350013. Available from: https://www.worldscientific.com/doi/full/10.1142/S146902682350013X

7. Cholakov R, Kolev T. Transformers predicting the future. Applying attention in next-frame and time series forecasting [Internet]. arXiv; 2021 [cited 2024 Mar 16]. Available from: http://arxiv.org/abs/2108.08224

8. Zhou H, Zhang S, Peng J, Zhang S, Li J, Xiong H, et al. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting [Internet]. arXiv; 2021 [cited 2024 Mar 7]. Available from: http://arxiv.org/abs/2012.07436

9. Elia: Belgian's Electricity System Operator [Internet]. Elia. [cited 2024 Mar 30]. Available from: https://www.elia.be/en/

10. Solar-PV power generation data [Internet]. [cited 2024 Mar 23]. Available from: https://www.elia.be/en/grid-data/power-generation/solar-pv-power-generation-data

11. Hecke AV, Lescrauwaet A, Verschelde J. Avhou/dne [Internet]. 2024. Available from: https://github.com/avhou/dne

Appendix A : Data general properties

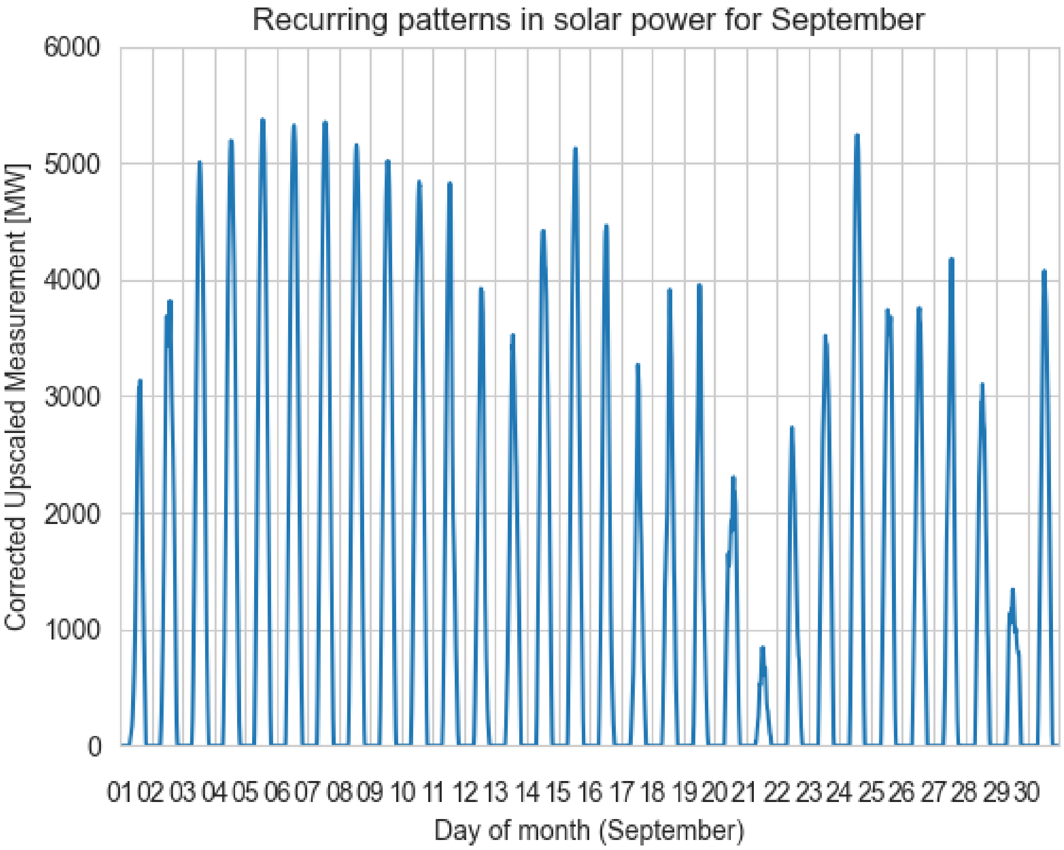

Figure 1: Typical recurrent patterns, here for September 2023

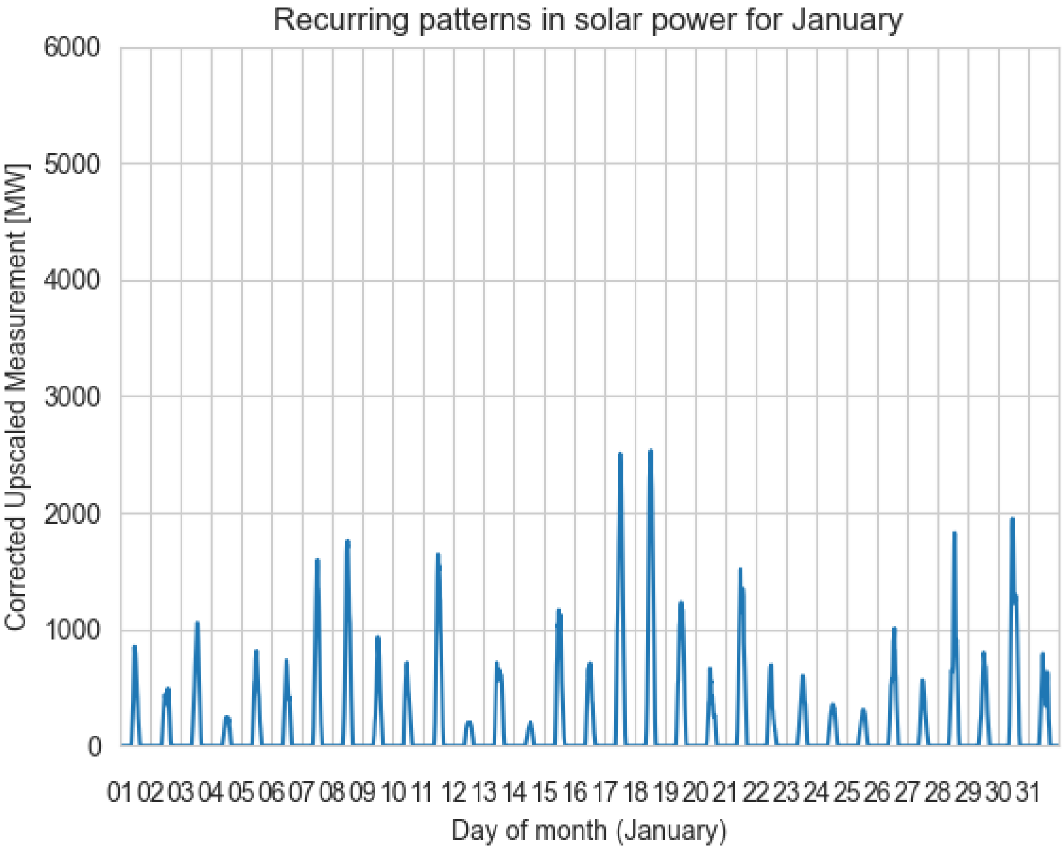

Figure 2: Typical recurrent patterns, here for January 2023