IM1102-232433M - Deep Neural Engingeering assignment 2

Modifying the attention mechanism of transformers for time series forecasting

Arne Lescrauwaet (852617312) - Joachim Verschelde (TODO) - Alexander Van Hecke (852631385)

March 30, 2024

Introduction

This report details the steps taken by Arne Lescrauwaet (852617312), Joachim Verschelde (TODO) 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, TODO, TODO, 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.

TODO iets zeggen over welke maanden we selecteren?

-> verschillende scenario's : - scenario 1 : zelfde maand (augustus) over alle jaren heen - scenario 2 : 3 maanden van een seizoen (zomer) voor 1 jaar - scenario 3 : zelfde maand (december) over alle jaren heen - scenario 4 : 3 maanden van een seizoen (winter) voor 1 jaar

te variëren met input embedding size 5 - 10 - 20 dagen?

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]

TODO extra features / embedding -> feature + positional encoding + one-hot encoding van dag of maand of week "temporal encoding"

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 scenario's 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 mechanims 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. 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

TODO op welke lengte gaan we onze input datasets opsplitsen

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 what the influence on prediction MSE was. Models were tuned using appropriate hyperparameters using 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. All analysis was done using the pytorch python package.

TODO beschrijven hoe split training / validation / test set.

Design elaboration

We decided to evaluate the following attention mechanisms :

- regular self-attention
- convoluted self-attention as described in (3)
- asymetric convoluted self-attention
- XYZ

-> kijken naar RCNN? -> eigenvectoren?

TODO experiment beschrijven naar dataset of datasets, parameters (bv kernel size) en hyperparameters (indien van toepassing)

hyperparameters : -> kernel size -> attention head (softmax tussen key en query) -> gridsearch, random search of half random search (scikit learn)?

Table 3: parameters used for the attention mechanisms

algo	parameter	range
regular self-attention	?	?

algo	parameter	range
convoluted self-attention	kernel size	2-10
XYZ	?	?

TODO indien meerdere scenarios (bv meerdere datasets, of een scenario zomer/winter/. . .) hier opsommen van scenarios

This entire design is repeated for a number of different scenarios. We detail these in Table 4.

Table 4: learning scenarios

scenario	description
summer	TODO july - september
winter	TODO december - february

Implementation

All code is available in a github repository (11).

TODO eventueel speciale vermeldingen rond implementatie

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 : Regular self-attention

This is the baseline scenario. Results are summarized in Table 5.

Table 5: one sample t-test to determine whether TODO

metric	metric mean	population mean	t-test value	p-value	H_0 rejected
accuracy	0.9580	0.9864	-1.77E+01	2.29E-32	yes
accuracy	0.9858	0.9864	-1.84E+01	1.17E-33	yes
accuracy	0.9782	0.9864	-2.58E+01	9.62E-46	yes

TODO resultaten beschrijven

Scenario 2 : Convoluted self-attention

TODO idem hierboven

Recap

TODO hier summary van scenarios

Table 6: Features importance summary

scenario	attention mechanism	TODO
summer	regular self-attention	?
summer	convoluted self-attention	?
winter	regular self-attention	?
winter	convoluted self-attention	?

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
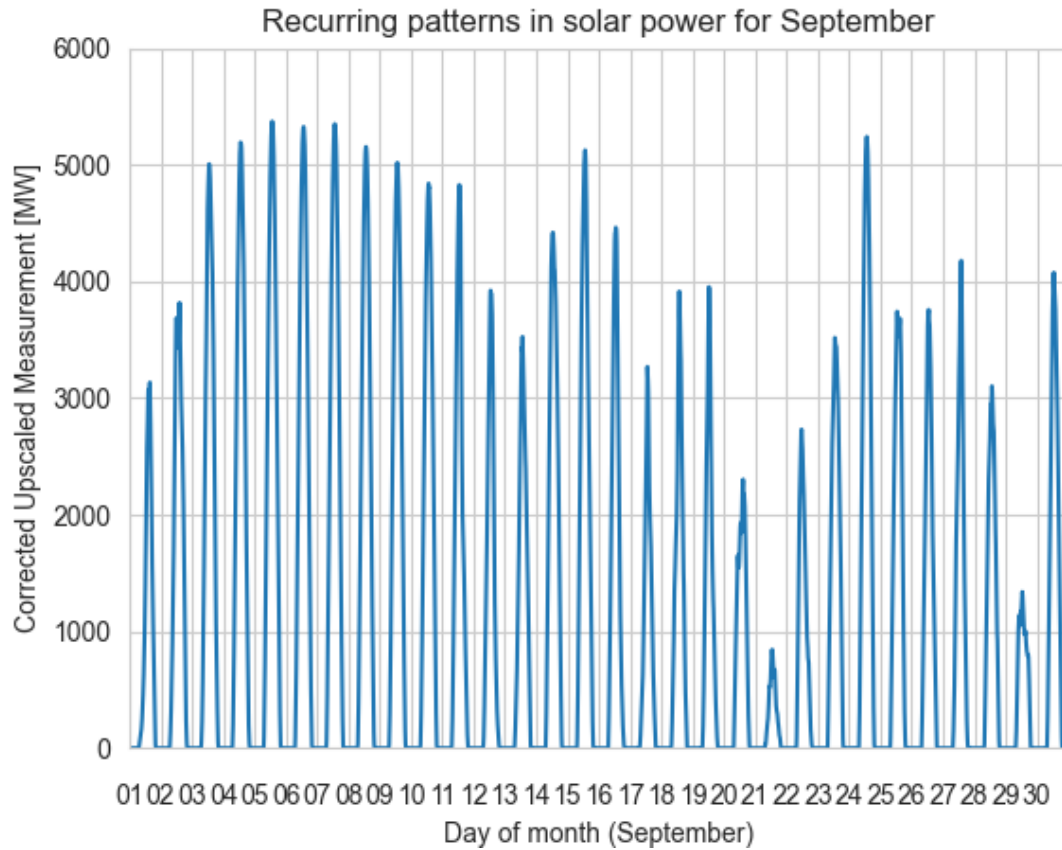
Appendix A : Data general properties

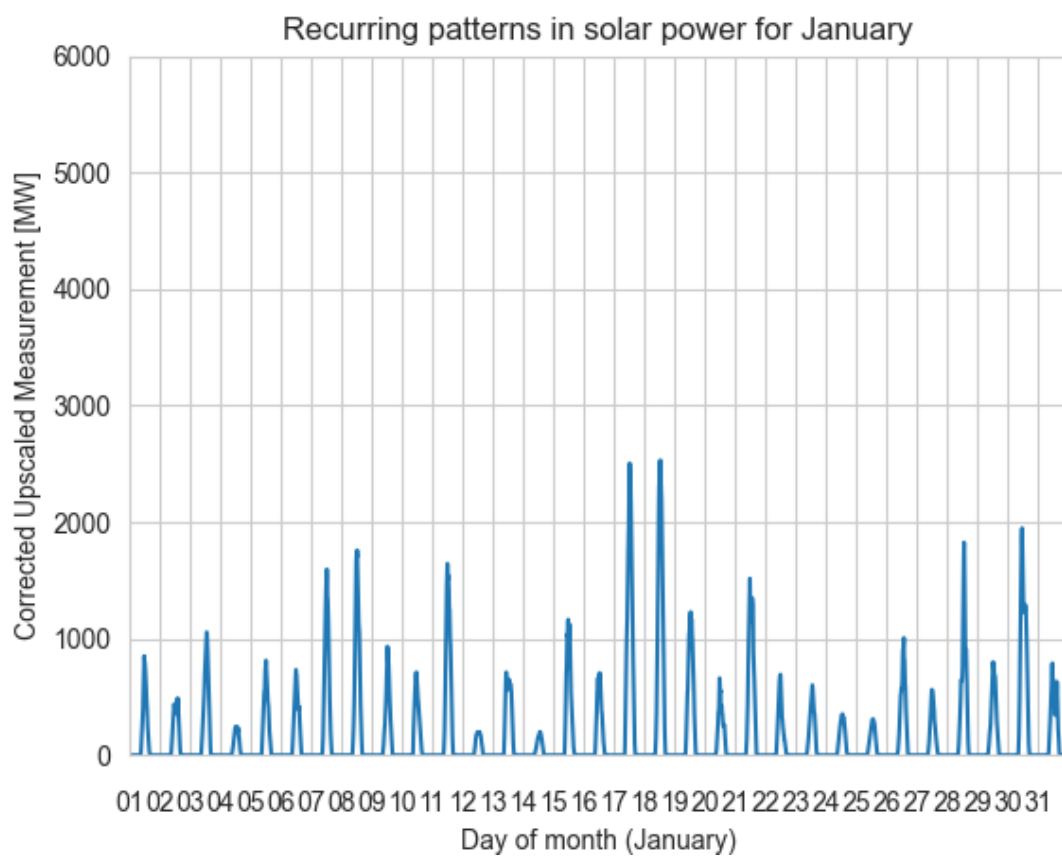Figure 1: Typical recurrent patterns, here for September 2023

Figure 2: Typical recurrent patterns, here for January 2023