Time series PF coil temperature forecasting using recurrent transformer model.

Giil Kwon, Taehyun Tak, Hyunjung Lee
KSTAR superconducting coil system

- The superconducting coil system is one of the most important components in Korea Superconducting Tokamak Advanced Research (KSTAR), which has been operated since 2008.
- Superconducting coils are cooled by forced-flow supercritical helium about 4.5 K.
- To protect the superconducting coil system, we need to predict next superconducting coil temperature.
- The rising of poloidal field (PF) coil’s temperature is mostly by AC losses according to the variations of current and magnetic field.
The measurement of the coil temperature

- Coil’s temperature are measured on Tokamak Monitoring System (TMS).
  - TMS measured the temperature data and publish PV every 1 second.

- Thermometer measure the temperature of coolant (Helium).
  - Thermometers are located at inlet and outlet of each cooling channels.
  - In this work, we use data from outlet channel 5 thermometer of the PF1 coil.
Problem formulation

- Suppose we have a collection of $N$ univariate time series data, $\{X_{n,t_0:t_j}\}_{n=1}^N$, where $X_{n,t_0:t_j} = [X_{n,t_0}, X_{n,t_1}, \ldots, X_{n,t_j}]$, $X_{n,t} \in \mathbb{R}$, denotes $n'th$ time series data value at time $t$. We will predict the next $\tau$ step time series values, $\{X_{n,t_{j+1}:t_{j+\tau}}\}_{n=1}^N$.
  - In this work, $N=1$, $\tau=1$, we will do one step ahead forecast of univariate time series data.
  - We will forecast the temperature of PF1L(channel 5) data.

- We are going to formulate the multi step prediction into one-step ahead prediction problem, where $\Phi$ is learnable parameter while training.

$$ P(X_{n,t_i+\Delta+1:t_i+\Delta+\tau+1}|X_{n,t_i:t_i+\Delta}; \Phi) = \prod_{t=t_i}^{t_i+\tau} P(X_{n,t+\Delta+1}|X_{n,t:t+\Delta}; \Phi) $$
Problem formulation

- We change this model to recurrent model by adding latent variable $Z_{n,t_i}$.
- $Z_{n,t_i}$ denotes the latent variable of $n^\text{th}$ time series data at time $t_i$.

$$P\left(\frac{X_{n,t_i+\Delta+1:t_i+\Delta+\tau+1}}{t_i+\tau} \mid X_{n,t_i:t_i+\Delta}; \Phi'\right)$$

\[= \prod_{t=t_i} P\left(\frac{X_{n,t+1:t+\Delta+1}}{t+\Delta}, Z_{n,t_i+1} \mid X_{n,t:t+\Delta}, Z_{n,t_i}; \Phi'\right)\]
Model Architecture

- **Long-Short-Term Memory (LSTM)**
  - LSTM is one kind of Recurrent neural network.
  - LSTM is well suited to predict, classify the time series data.

- **Transformer**
  - Transformer is the self-attention based deep learning model.
  - It mostly used to solve Natural Language Problem (NLP) (such as GPT-3, BERT).
  - Nowadays, it also used to forecast time series data (Informer, Perceiver, Reformer)

- **Perceiver**
  - The model build upon the Transformer that can learn multimodal data.
  - The model iteratively attend to the input data by alternating cross-attention and latent transformer.
  - The model unrolled in depth to the same input rather then in time to different inputs.
Model Architecture

- **Recurrent Transformer (RT)**
  - We build recurrent transformer upon the perceiver by iteratively giving time series input and attend input data to latent variable by using cross-transformer module.
  - As the perceiver did, our model also alternating cross-attend module and latent transformer.
  - As RNN did, we give latent variable as input to the model.
  - Unlike Perceiver, our model unrolled in time to different inputs.
Model Architecture

**Recurrent Transformer**

![Diagram of Recurrent Transformer](image)

Cross Attention

\[
\text{Cross Attention}(Q', K, V) = \text{softmax} \left( \frac{Q' K^T}{\sqrt{d_k}} \right) V,
\]

where \( Q' \in \mathbb{R}^{\Delta x D_L}, K \in \mathbb{R}^{\Delta x D_I} \) and \( V \in \mathbb{R}^{\Delta x D_I} \)

\( D_I \) denotes dimension of input data (1).
\( D_i \) indicates dimension of latent variable (48).
\( D_I \ll D_L \)
Model Architecture

❖ Recurrent Transformer

Input($X_{n,t_i:t_i+\Delta}$)

Latent($Z_{n,t_i}$)

Latent'($\Delta x D_L$)

Latent($\Delta x D_L$)

Output($X_{n,t+\Delta+1}$)
KSTAR Dataset

- Data acquired from Magnet Power Supply (MPS), Helium Distribution System (HDS), Tokamak Monitoring System (TMS).
  - MPS generate PF current related PV, TMS generate temperature related PV, HDS generate helium pressure and flow rate related PV.
- We have interpolated the data to have each data every 1 seconds. and remove noise.
- It consists of data collected between 2018/11/01 and 2018/11/30 (633 shots, 24 days, 820800 samples)
  - Train data (19 days, 615600 samples) : 79% 
  - Test data (5 days, 171000 samples): 21%
Experiment

Environment

- We use Pytorch library to build our program.
- We use 1 GPU card when we training the dataset.

<table>
<thead>
<tr>
<th>Hardware specification</th>
<th>Name</th>
<th># of HW</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>AMD Ryzen Threadripper 1920x</td>
<td>2</td>
</tr>
<tr>
<td>GPU</td>
<td>Nvidia 2070 super</td>
<td>1</td>
</tr>
</tbody>
</table>

Dataset
Training

- We used Adam W optimizer with $\beta_1 = 0.9, \beta_2 = 0.99$ and $\epsilon = 10^{-8}$. And StepLR scheduler with step size = 1.0 and $\gamma = 0.9$.
- Model dimension = 48 and latent dimension = 48
- Training Time: it takes 20 hour to train the Recurrent Transformer that uses 100 length sequence as input (256 batch).

Model Parameter Setting

- LSTM: hidden size = 96, # of layer = 1, Epoch = 200, batch size = 256, input window = 100, input dimension = 47
- Transformer: # of head = 8, # of layer = 2, Epoch = 200, batch size = 256, input window = 100, input dimension = 48
- Recurrent Transformer: # of head = 8, # of layer = 1, Epoch = 200, batch size = 256, input window = 100, input dimension = 48
Experiment

Forecast graph (total test set result)

- Transformer
- LSTM
- Recurrent Transformer (seq1)
- Recurrent Transformer (seq100)
Experiment

- Forecast graph (one shot result)

13th Technical Meeting on Plasma Control Systems, Data Management and Remote Experiments in Fusion Research
Experiment

- Forecast graph (one shot result)

![Graphs showing forecast results for different models: Transformer, LSTM, Recurrent Transformer (seq1), Recurrent Transformer (seq100).]
One step ahead forecast test result

- We compare forecast accuracy of the Recursive Transformer (RT) with other algorithms (LSTM, Transformer) which has similar parameters (batch size = 256, model dimension = 48, # of epoch = 200).
  - RT with sequence length 100 (seq100) is more accurate than others
  - RT with sequence length 1 (seq1) also has similar forecast accuracy to the conventional Transformer.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>R2 score</th>
<th>MAE</th>
<th>RMSE</th>
<th>MAPE</th>
<th>Quantile score (P10/P50/P90)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>0.9916</td>
<td>0.0048</td>
<td>0.0198</td>
<td>0.0377</td>
<td>(0.0024/0.0024/0.0024)</td>
</tr>
<tr>
<td>Transformer (2 layer, seq 100)</td>
<td>0.9992</td>
<td>0.0029</td>
<td>0.0061</td>
<td>0.0830</td>
<td>(0.0019/0.0014/0.0009)</td>
</tr>
<tr>
<td>Recursive Transformer (seq1)</td>
<td>0.9973</td>
<td>0.0017</td>
<td>0.0111</td>
<td>0.0258</td>
<td>(0.0009/0.0008/0.0007)</td>
</tr>
<tr>
<td>Recursive Transformer (seq64)</td>
<td>0.9988</td>
<td>0.0018</td>
<td>0.0071</td>
<td>0.0892</td>
<td>(0.0014/0.0009/0.0003)</td>
</tr>
<tr>
<td>Recursive Transformer (seq100)</td>
<td>0.9992</td>
<td><strong>0.0013</strong></td>
<td><strong>0.0061</strong></td>
<td><strong>0.0112</strong></td>
<td><em>(0.0009/0.0006/0.0003)</em></td>
</tr>
</tbody>
</table>
Experiment

- Inference time
  - We measured one sample inference time of models.
  - The inference time of RT when seq1 is about 3.2541 ms.
  - The inference time of RT when sequence length 100 is about 21.43377 ms.
  - The inference time of Transformer when sequence length 100 is about 19.78277 ms.
  - The inference time of RT when sequence length 100 is slightly larger than the transformer inference time.
Conclusion

- We have presented the Recurrent Transformer, a perceiver based model that can one step ahead forecast of univariate time series data.
  - Our model successfully learn and one step ahead forecast of KSTAR PF1 temperature data.
  - Our model predicted time series data more accurately than other existing models (LSTM, Transformer).

- In the future, we would like to forecast multi step ahead forecast using univariate data or using multi variate data.
The End
Appendix
Why we choose channel 5 of PF 1 temperature sensor

- To measure the temperature variation caused by AC loss, we chose the channel 5.
- because Channel 5 is most affected by magnetic field
  =>This channel mostly affected by operation of PF coil

Why temperature of PF1 coil

- We have already recognized that PF1 coil shows a larger temperature rise compared to other PF coils and it will become one of important factors for KSTAR PF coil operation.
Performance Metrics

- R2-score
  
  Suppose $y_1, y_2, \cdots, y_n$ denote dataset values, $f_1, f_2, \cdots, f_n$ denote predicted values, and $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$
  
  $$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \bar{y})^2}{\sum_{i=1}^{n} (y_i - f_i)^2}$$
  
  Best score is 1.0 and worst score is 0.0

- Mean Absolute Error (MAE)
  
  $$MAE = \frac{\sum_{i=1}^{n} (y_i - f_i)^2}{n}$$

- Root Mean Square Error (RMSE)
  
  $$RMSE = \sqrt{MSE} = \sqrt{\frac{\sum_{i=1}^{n} (y_i - f_i)^2}{n}}$$

- Mean Absolute Percentage Error (MAPE)
  
  $$MAPE = \frac{100}{n} \sum_{i=1}^{n} \frac{|y_i - f_i|}{y_i}$$

- Quantile
  
  Quantile indicates the area below the normal distribution curve of error between $y_i$ and $f_i$.
  
  The (P10/P50/P90) quantile represents values with a loss value of less than (10%/50%/90%).