CS 550 Machine Learning

Paper Review: Sequence Prediction Using Spectral RNNs

Emirhan Koç
Bilkent University
Electrical and Electronics Engineering
emirhan.koc@bilkent.edu.tr

***Abstract*—Computation cost and accuracy are two essential component in assesment of model performance in machine learning and deep learning applications. Recently, use of Fourier Transform and its variants attracted attention in deep learning applications and showed remarkable enhancement in terms of computational complexity without significant loss of accuracy. In this paper, a novel approach in sequence prediction is proposed that combines short-time Fourier transform and recurrent neural networks. Besides the experiments shared in the original paper, several experiments are performed on financial data and artificially produced data which created using first order auto-regressive process and nonlinear auto-regressive moving average processes.**

***Index Terms*—Sequence Prediction, Short-Time Fourier Transform, Recurrent Neural Networks**

I. Introduction

The signals used in various applications are mostly raw and unstructured form so that it requires meticulous analysis and processing to extract meaningful information in it. In addition, mapping between domains such as from time to frequency or space to frequency gives power of insight and interpretation to acquire satisfying results in corresponding applications. There are several transformation that is used in signal processing applications such as Fourier [1], Short-Time Fourier Transform [2], Wavelet Transform and Hilbert Transform. [3] Recently, application of these transforms or their combinational use has been drawn attention in machine learning and deep learning research. Since these transforms are standard, non-parameterized and linear, lately, they are in the spotlight in deep learning applications. In this paper, they showed that gradient can be propagated through the STFT and its inverse. This enabled the use of time-domain signal in frequency domain, process this complex-valued signal through recurrent neural networks and calculate loss in time domain after inverse transform. It is also proposed that filtering the signal in frequency domain may be beneficial to reduce the effect of undesired patterns such as noise in the signal. In addition, this filtering operation allows to reduction in the size of input without significant loss in the representation in the signal. Yet, working with Fourier transformed signals requires handling complex-valued signals and different approaches are proposed to handle this challenge.

II. Related works

Recently, Fourier and Short-Time Fourier Transforms have been started to used in image processing and vision applications. One interesting study is **Fourier Convolutional Neural Networks** and the findings of this study shows how computational complexity decreases drastically. [4] The idea in this study comes from basic property of Fourier Transform that convolution in spatial domain corresponds to multiplication in frequency domain. In this study, it is claimed that replacing convolution operation in convolution layer with Hadamard multiplication [5] yields similar performance in terms classification accuracy with huge decline in computational cost. In order to implement this idea, both inputs (images) are converted into frequency domain and kernels (weights) are initialized in frequency domain.As it is stated above, without significant loss of classification correctness, cost of computation diminishes with the use of frequency domain. [4] A brand new study, **FNet: Mixing Tokens with Fourier Transform**, showed that in deed self-attention layer in encoder block brings about large complexity and non-linearity. In this study, since it is linear, unparameterized and in standard form, self-attention layer can be replaced by Fourier Transform. It is worth to note that it is only restricted to encoder block as part of this study. Surprisingly, results of experiments on several well-studied benchmark dataset exhibit substantial decline in cost of computation on TPU and GPU with reasonable amount of loss in accuracy. It is also shown that novel architecture surpasses more previous model with increasing input size. [6]

III. Methodology

Time series forecasting is a common field of application as it is used in variety of areas such as weather prediction, sensors data and finance. [7] Since they are studied well and their power is proved on sequential data analysis, RNNs, LSTM and GRU are principally used in sequence prediction tasks. [8]Likewise applications mentioned above, Short-Time Fourier Transform is used in sequence prediction in the study **Sequence Prediction Using Spectral RNNs** which greatly deals with backpropagation in complex domain, additionally. In this study, using a Gaussian window, time series is divided into segments and Fourier transform is applied to each segment. In other words, time series is converted to frequency domain using Short-Time Fourier Transform. Unlike above studies, frequency series are used, not time-frequency maps.

As the network architecture , GRUs with several hidden unit sizes are used and experiment results are presented. As data in frequency domain is complex valued, they concatenate absolute of complex part to real part. This brings about the increase in parameter count as the size of hidden state is doubled. In addition, they offered a GRU architecture which is called complex gated recurrent units that deals greatly with backpropagation in complex domain. According to presented results, prediction loss using transformed data is less than loss of time domain data. Considering results on transformed data, it is also observed that increase hidden unit size affect significantly the prediction loss such that bigger architecture yields better results up to 3 times smaller prediction loss. For same tasks, models with complex gated recurrent units outperform results obtained by common GRUs. [9]

A. Short-Time Fourier Transform

Short Time Fourier Transform (STFT) is a special variant of common Fourier Transform that is extensively used in time series analysis [5]. It is wisely to use STFT in time-variant and long signal analysis since the non-stationary characteristics of the signal may yield improper insight on data. Instead of taking the signal as a whole, the signal can be sliced into sub-segments via a windowing function. It is commonly Gaussian [10] and Hanning [11] windows. Afterward, same procedure of Fourier Transform applied each segment separately.

$$X(S_m, w) = \sum_{n=-\infty}^{\infty} x_n w_{n-m} e^{-jwn} \tag{1}$$

where w_n is the window function. For instance, the Gaussian window shown below:

$$w_n = e^{\frac{-1}{2} * (\frac{n - \frac{T}{2}}{\sigma \frac{T}{2}})^2} \tag{2}$$

In Eq.1, S_m represents the m'th segment of the signal after dividing into segments.

B. Complex Spectral Recurrent Neural Network Architecture

Initially, the input signal $\mathbf{X}$ is converted into the its frequency domain representation using STFT and processed using RNNs. After processing of the frequency domain signal, resulting output $\mathbf{Y}$ is converted back to the time domain using iSTFT to calculate loss and gradient values. In general, network architecture can be summarized in Fig. 1. [9]

$$\mathbf{X}_\tau = \mathcal{F}(\{\mathbf{x}_{S_\tau - T/2}, \ldots, \mathbf{x}_{S_\tau + T/2}\}) \tag{3}$$
$$\mathbf{z}_\tau = \mathbf{W}_c \mathbf{h}_{\tau-1} + \mathbf{V}_c \mathbf{X}_\tau + \mathbf{b}_c \tag{4}$$
$$\mathbf{h}_\tau = f_a(\mathbf{z}_\tau) \tag{5}$$
$$\mathbf{y}_\tau = \mathcal{F}^{-1}(\{\mathbf{W}_{pc}\mathbf{h}_0, \ldots, \mathbf{W}_{pc}\mathbf{h}_\tau\}) \tag{6}$$

where $\tau = [0, n_s]$, *i.e.* from zero to the total number of segments n_s. The output y_τ may be computed based on the available outputs $\{\mathbf{W}_p\mathbf{h}_0, \ldots, \mathbf{W}_p\mathbf{h}_\tau\}$ at step τ. Adding the STFT-iSTFT pair has two key implications. First of all, because $\mathbf{X}_\tau \in \mathbb{C}^{n_f \times 1}$ is a complex signal, the hidden state as well as subsequent weight matrices all become complex, *i.e.* $\mathbf{h}_\tau \in \mathbb{C}^{n_h \times 1}$, $\mathbf{W}_c \in \mathbb{C}^{n_h \times n_h}$, $\mathbf{V}_c \in \mathbb{C}^{n_h \times n_f}$, $\mathbf{b}_c \in \mathbb{C}^{n_h \times 1}$ and $\mathbf{W}_{pc} \in \mathbb{C}^{n_h \times n_f}$, where n_h is the hidden size of the network as before and n_f is the number of frequencies in the STFT.

Fig. 1: Here each input X_{S_τ} denotes an n_f dimensional vector where n_f is the total number of frequencies and X_τ is the transformed version of them for each segments seperately. Then, each of them is fed into RNNs and outputs are calculated. At last stage, resulting signal of the last cell is converted back to time domain with inverse STFT to compute loss and gradient.

IV. Experiments and Results

A. Experiment in Paper

In this study, a synthetic time series is created using Mackey-Glass Equation which is a nonlinear time delay differential equation. This synthetic data is created with the given equation below. [12]

$$\frac{dx}{dt} = \frac{\beta x_\tau}{1 + x_\tau^n} - \gamma$$

where x with $\beta : 0.2$, $\gamma : 0.1$, $\tau = 17$, t= [0,512]

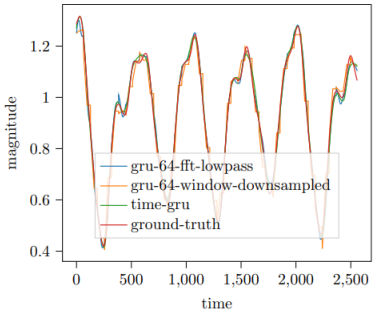

Fig. 2: Mackey-Glass series predictions in for different RNN methods. Note: This is the results on test set

TABLE I: Results on Mackey-Glass Data

Architecture	MSE	Time
time-GRU	$3.8 * 10^{-4}$	355
STFT-GRU	$3.5 * 10^{-4}$	57
STFT-cgRNN-32	$2.1 * 10^{-4}$	63
STFT-cgRNN-64	$1.1 * 10^{-4}$	63
STFT-cgRNN-64$_{L_C}$	$210 * 10^{-4}$	64

TABLE II: For example in STFT-cgRNN-32, cgRNN denotes complex gated RNNs are used and numbers at rightmost represents hidden unit size, and L_C denotes loss is calculated in frequency domain.

Here, it is showed that GRU with FFT operation result in decrease in loss, complex gates results in more decrease in loss but loss calculation in frequency domain (inverse STFT is not calculated) is not promising compared to other architectures.

B. New Experiments

I performed two different sets of experiment with currency data and synthetically created data using random process equations.

1) USD-TRY and EU-USD Time Series Prediction: Here, I collected data starting from January 2011 to January 2021 in daily resolution. [13] Information of some days were missing, instead of discarding them, I take the average of the day before and after that day.

The figures at bottom shows that their proposed models overfits the data and cannot make satisfying predictions a different data. In their proposed model, they iterate 30.000 times with less amount of data points. This also causes the model to overfit data. I am also suspicious that they somewhere use test data in their algorithm as the algorithm learns very hard patterns in test set.

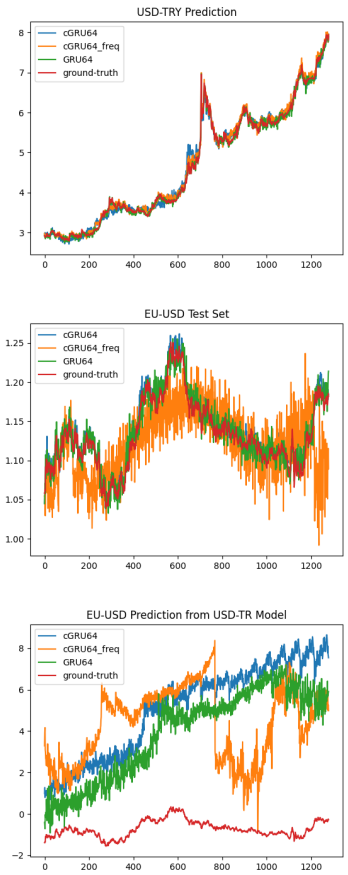

Fig. 3: Top:USD-TRY Prediction, Middle: EU-USD Prediction, Bottom: EU-USD Prediction using USD-TRY Model

TABLE III: Top: USD-TRY, Bottom: EU-USD

Architecture	MSE	Time
STFT-GRU	$2.24 * 10^{-5}$	3
STFT-cgRNN-32	$1.8 * 10^{-4}$	8
STFT-cgRNN-64	$5.42 * 10^{-5}$	11
STFT-cgRNN-64$_{L_C}$	$2.18 * 10^{-6}$	11
STFT-GRU	$1.33 * 10^{-4}$	1
STFT-cgRNN-32	$1.33 * 10^{-4}$	3
STFT-cgRNN-64	$1.09 * 10^{-4}$	4
STFT-cgRNN-64$_{L_C}$	$1.2 * 10^{-4}$	4

2) Synthetically Created Data: In this experiments, I used two different data that is artificially produced using random processes.

Nonlinear-Autoregressive Moving Average-10th Order:

y(t) = αy(t-1)+ βy(t-1) $\sum_{i=1}^{n} y(t-i)$ +γu(t-1)u(t-n)+ δ here α=0.3, β=0.05, δ=10, γ=1.5, n = 10 and u is uniformed random variable. The data is produced relying on this equation. [14]

The other random process is first order auto-regressive process with deterministic switch. The data is produced according to equation in Figure 4. [15]

$$x_{t+1} = \begin{cases} x_t + \epsilon & \text{if} \quad \text{mod } (t, 1000) < 500 \\ -0.9\, x_t + \epsilon & \text{if} \quad \text{mod } (t, 1000) \geq 500 \end{cases}$$

Fig. 4: AR(1) with deterministic switching with 2 regime

TABLE IV: Top: NARMA(10), Bottom: AR(1)

Architecture	MSE	Time
STFT-GRU	$3.85 * 10^{-5}$	1
STFT-cgRNN-32	$6.08 * 10^{-5}$	3
STFT-cgRNN-64	$6.51 * 10^{-5}$	3
STFT-cgRNN-64$_{L_C}$	$8.3 * 10^{-5}$	4
STFT-GRU	$6.7 * 10^{-4}$	1
STFT-cgRNN-32	$8.1 * 10^{-4}$	4
STFT-cgRNN-64	$7.08 * 10^{-4}$	4
STFT-cgRNN-64$_{L_C}$	$1.05 * 10^{-2}$	5

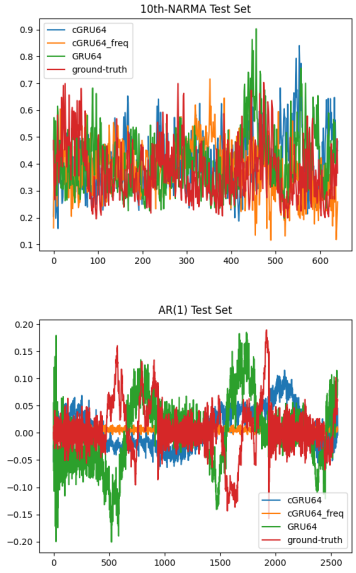

Fig. 5: Top:NARMA Prediction, Bottom: AR(1) Prediction

V. Conclusion and Discussion

According to proposed paper and its results, even it requires great effort to implement RNNs with complex value processing, they achieved to develop this novelty and furthermore, obtained satisfactory results. It was a comprehensive study that investigate not only effect of changing domain from time to frequency but also shows how filtering and cell size effect the results. Moreover, they showed how working with complex valued signals and complex valued weights bring about better results. There is also one interesting topic. As the initial input signals are real valued, their Fourier transforms will be complex-conjugate which means half of the signal in frequency domain can give us the other half. They used this property to reduce input dimension. This is a highly intellectual idea to consider and apply. Yet, when I produced results with the full length of spectrum, instead of the half, I obtained better results. It was interesting because they must

also be performed the same experiment and results must be similar to I obtained. I am curious about why they did not share this result.

REFERENCES

[1] M. Corinthios, "A generalized transform, grouping, fourier, laplace and z transforms," *The 14th International Conference on Microelectronics,*.

[2] L. Li, H. Cai, H. Han, Q. Jiang, and H. Ji, "Adaptive short-time fourier transform and synchrosqueezing transform for non-stationary signal separation," *Signal Processing*, vol. 166, p. 107231, 2020.

[3] J. Shi, Y. Zhao, W. Xiang, V. Monga, X. Liu, and R. Tao, "Deep scattering network with fractional wavelet transform," *IEEE Transactions on Signal Processing*, vol. 69, p. 4740–4757, 2021.

[4] H. Pratt, B. Williams, F. Coenen, and Y. Zheng, "Fcnn: Fourier convolutional neural networks," *Machine Learning and Knowledge Discovery in Databases*, p. 786–798, 2017.

[5] E. O. Brigham, *The fast fourier transform and its applications*. Prentice-Hall, 1988.

[6] J. Lee-Thorp, J. Ainslie, I. Eckstein, and S. Ontanon, "Fnet:mixing tokens with fourier transform," 2021.

[7] Z. Li, Y. Li, F. Yu, and D. Ge, "Adaptively weighted support vector regression for financial time series prediction," *2014 International Joint Conference on Neural Networks (IJCNN)*, 2014.

[8] S. Kumar, L. Hussain, S. Banarjee, and M. Reza, "Energy load forecasting using deep learning approach-lstm and gru in spark cluster," *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)*, 2018.

[9] M. Wolter, J. Gall, and A. Yao, "Sequence prediction using spectral rnns," in *29th International Conference on Artificial Neural Networks*, 2020.

[10] S.-C. Pei and S.-G. Huang, "Adaptive stft with chirp-modulated gaussian window," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[11] W. Yuegang, J. Shao, and X. Hongtao, "Non-stationary signals processing based on stft," *2007 8th International Conference on Electronic Measurement and Instruments*, 2007.

[12] M. Farzad, H. Tahersima, and H. Khaloozadeh, "Predicting the mackey glass chaotic time series using genetic algorithm," *2006 SICE-ICASE International Joint Conference*, 2006.

[13] "Yahoo finance - stock market live, quotes, business and finance news."

[14] A. Goudarzi, P. Banda, M. R. Lakin, C. Teuscher, and D. Stefanovic, "A comparative study of reservoir computing for temporal signal processing," 2014.

[15] F. Ilhan, O. Karaahmetoglu, I. Balaban, and S. S. Kozat, "Markovian rnn: An adaptive time series prediction network with hmm-based switching for nonstationary environments," *IEEE Transactions on Neural Networks and Learning Systems*, p. 1–14, 2021.