A DUAL-STAGE ATTENTION MECHANISM FOR TIME-SERIES PREDICTION

Bhushan D Deo, Saankhya S Mondal

IISc, Bengaluru, M.Tech AI

ABSTRACT

Next time step prediction in stock markets can lead to massive incentives to traders. This project starts with the Dual-attention RNN (DA-RNN)(1) baseline and builds upon it for next time step prediction on the NASDAQ100 time series. Exogenous(driving) time series output is utilized to effectively predict the future using the past. Our contributions include architecture changes, multi-step prediction and recasting the Transformer(2) architecture along with a modified time2vec(3) encoding.

1. INTRODUCTION

Predicting future values of the time series based on past values and available values for exogenous(driving) time series data is usually studied using Nonlinear autoregressive exogenous models (NARX). In the well cited paper from (1), the authors combat the long term dependency issues associated with the above models and use a dual attention scheme for doing future step prediction.

2. TECHNICAL DETAILS

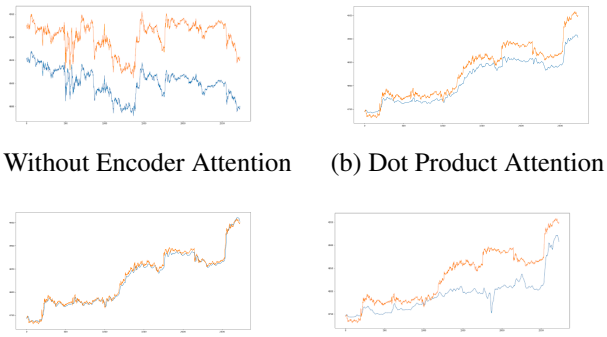
The authors argue for a dual-attention scheme inspired from behavioral mechanics both in the Encoder and the Decoder models.

2.1. Data

The input data at hand is $\boldsymbol{x_t^i}$. Here $i \in \{1, 2, ..., N\}$ and $t \in \{1, 2, ..., T\}$. Each $\boldsymbol{x_t^i}$ thus denotes the i^{th} exogenous time series, sliced at time instant t. We also have the history $y_h(t), t \in 1, 2, ..., T$, a single dimensional time series. Given this information, we have to predict $y_h(t), t \in T+1, ..., T+k$, where k is the prediction horizon. The paper deals with $k = 1$, as a part of a new contribution we also experiment with $k = 3$.

2.2. Attention Encoder

An attention mechanism is used to feed in the modified input at time t, $\tilde{\mathbf{x}}_t = \left(\alpha_t^1 x_t^1, \alpha_t^2 x_t^2, \cdots, \alpha_t^n x_t^n\right)^\top$. The coefficients α_t^k are softmax probabilities derived from a similarity measure between h_{t-1} and x_t^k.



(a) Without Encoder Attention (b) Dot Product Attention

(c) All previous hidden states (d) Baseline

Fig. 1. Initial Starting points for various models

2.3. Attention Decoder

The architecture is very similar to a standard Decoder architecture, except that the context vector is $\mathbf{c}_t = \sum_{i=1}^{T} \beta_t^i \mathbf{h}_i$. The coefficients β_t^k are softmax probabilities derived from a similarity measure between d_{t-1} and h_t.

3. RESULTS

We used the source code from (4) to implement our baseline results. The details are in the comparison section below.

4. CONTRIBUTIONS

Firstly we verified the need for dual attention. We concluded that dual attention is indeed necessary, otherwise the model is unable to learn the bias in the time series as shown in figure(a). We tried using scaled dot product attention instead of the attention scheme used in the paper. Specifically we replaced $e_t^k = \mathbf{v}_e^\top \tanh\left(\mathbf{W}_e\left[\mathbf{h}_{t-1}; \mathbf{s}_{t-1}\right] + \mathbf{U}_e \mathbf{x}^k\right)$ with $e_t^k = \left(\mathbf{W}_e\left[\mathbf{h}_{t-1}; \mathbf{s}_{t-1}\right]\right)^T \mathbf{U}_e \mathbf{x}^k$ and exactly the same on the Decoder side. This resulted in lesser initial error as compared to the baseline and better learnability, figure(b). Our second contribution includes using all the previous hidden decoder states to predict the output. In particular we change $\hat{y}_T = \mathbf{v}_y^\top \left(\mathbf{W}_y\left[\mathbf{d}_T; \mathbf{c}_T\right] + \mathbf{b}_w\right) + b_v$ to $\hat{y}_T = \mathbf{v}_y^\top \left(\mathbf{W}_y\left[\mathbf{d}_T^{net}; \mathbf{c}_T\right] + \mathbf{b}_w\right) + b_v$, where $\mathbf{d}_T^{net} =$

$F(\mathbf{d}_1, ..., \mathbf{d}_T)$. We simply use a MLP for F. Our third contribution involves using multi-head attention scheme instead of using just one head. Our final experiment is to use a Transformer for the predictions. All the reported plots show initial response of the models subject to random initialization. The final results are omitted in order to maintain brevity and clarity. We have included them in the presentation.

5. RESOURCES

We used pytorch as our base DL library. All the experiments were run on a **RTX 8000 Quadro** GPU.

6. REFERENCES

[1] Qin, Yao, et al. "A dual-stage attention-based recurrent neural network for time series prediction." arXiv preprint arXiv:1704.02971 (2017).
[2] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.
[3] Time2Vec for Time Series features encoding https://towardsdatascience.com/time2vec-for-time-series-features-encoding-a03a4f3f937e
[4] https://github.com/KurochkinAlexey/DA-RNN