Autoregressive Convolutional Neural Networks for Asynchronous Time Series

11/29/2018

Authors: Mikołaj Binkowski, Gautier Marti, Philippe Donnat

Presented by: Junyi Zhang

STAT946 Deep Learning (Fall 2018)

UNIVERSITY OF WATERLOO

Introduction

- Significance-Offset Convolutional Neural Network for Asynchronous time series
- Combining AR models and Neural Networks
- Inspired by standard AR models and gating mechanisms (used in RNN)
- Focused on time series with multivariate and noisy signals i.e. financial data
- Time series forecasting problem:

$$p(X_{t+d}|X_t, X_{t-1}, ...) = f(X_t, X_{t-1}, ...)$$

Introduction

- Financial time series is challenging to predict due to their low signal-to-noise ratio and heavy-tailed distributions
  - same signal (e.g. price of a stock) obtained from different sources (e.g. financial news) asynchronously, each source may have different bias or noise (Fig1)
  - the traditional econometric models i.e. AR,VAR,VARMA might not be sufficient.
  - combine them with deep neural networks that can learn highly nonlinear relationships

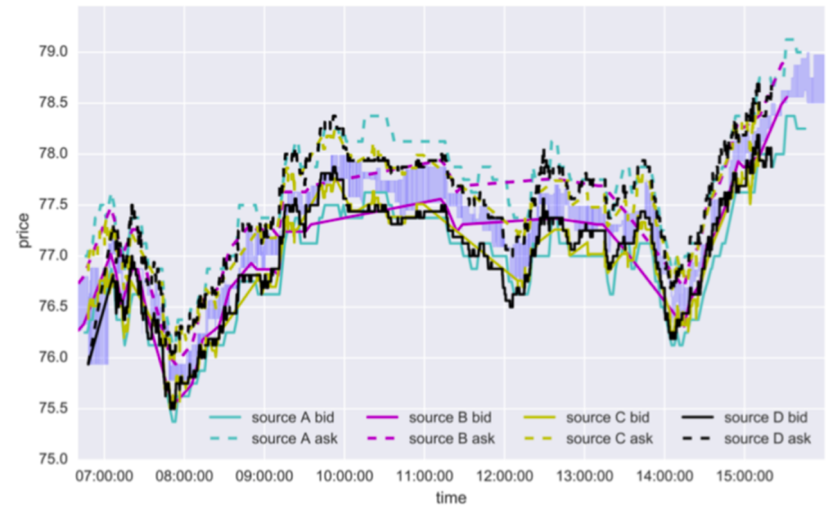


Figure 1. Quotes from four different market participants (sources) for the same CDS[2] throughout one day. Each trader displays from time to time the prices for which he offers to buy (*bid*) and sell (*ask*) the underlying CDS. The filled area marks the difference between the best sell and buy offers (*spread*) at each time.

UNIVERSITY OF WATERLOO

Related Work

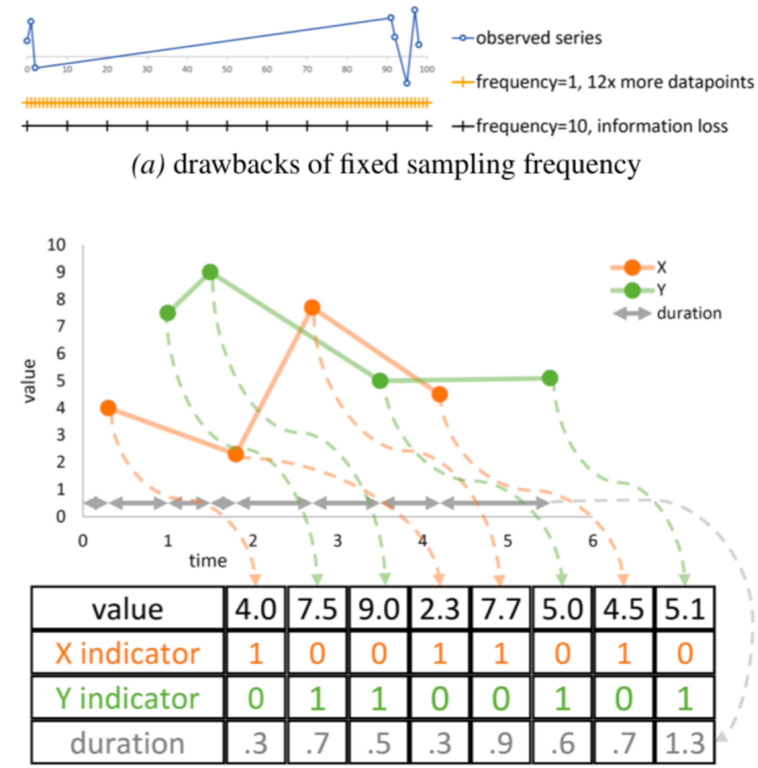
- Stochastic models such as AR, ARIMA and GARCH
  - Explain variables rather than improving out-of-sample prediction
  - Overfit, poor out-of-sample performance
- Gaussian processes, especially irregular sampled time series
  - Combine with econometric models i.e. Gaussian Copula Process Volatility[1]
- 4-layer perceptrons in modeling price change distributions in Limit Order Books[2]
- WaveNet architecture to several short univariate and bivariate time-series (including financial ones)[3]
- Autoencoders with a single hidden layer to compress multivariate financial data.[4]
- Neil et al. (2016): Augmentation of LSTM architecture suitable for asynchronous series[5], which stimulates learning dependencies of different frequencies through time gate.

Related Work

- **Gating and weighting mechanisms**
  - overcome the problem of vanishing gradient
  - $f(x) = c(x) \otimes \sigma(x)$
  - f(x): output function, c: nonlinear function of x, $\otimes$: an element-wise(Hadamard) matrix product, $\sigma(x)$: a sigmoid nonlinearity that controls the amount of output passed to the next layer
  - Appropriate composition of functions may lead to popular recurrent architecture such as LSTM and GRU.

Motivation

- Econometrics and machine learning communities done independently. AR models are not sufficient.
- Gaussian processes follow heavy-tailed distribution for financial datasets.
- Irregular sampled time series involve highly nonlinear functions.
- The dimension of multivariate time series are often observed separately and asynchronously, fix frequency may lead to lose information or enlarge the dataset (Fig2).
  - Treats varying durations as additional feature
  - uses the indicator feature to indicate whether the value of the observation is in this duration or not.
- LSTM increases computation complexity.



(a) drawbacks of fixed sampling frequency

(b) representation

Figure 2. (a) Fixed sampling frequency and its drawbacks; keeping all available information leads to much more datapoints. (b) Proposed data representation for the asynchronous series. Consecutive observations are stored together as a single *value* series, regardless of which series they belong to; this information, however, is stored in *indicator* features, alongside durations between observations.

UNIVERSITY OF WATERLOO

Model Architecture

- Given multivariate time series $(x_n)_{n=0}^{\infty} \subset \mathbb{R}^d$
- Predict the conditional future values

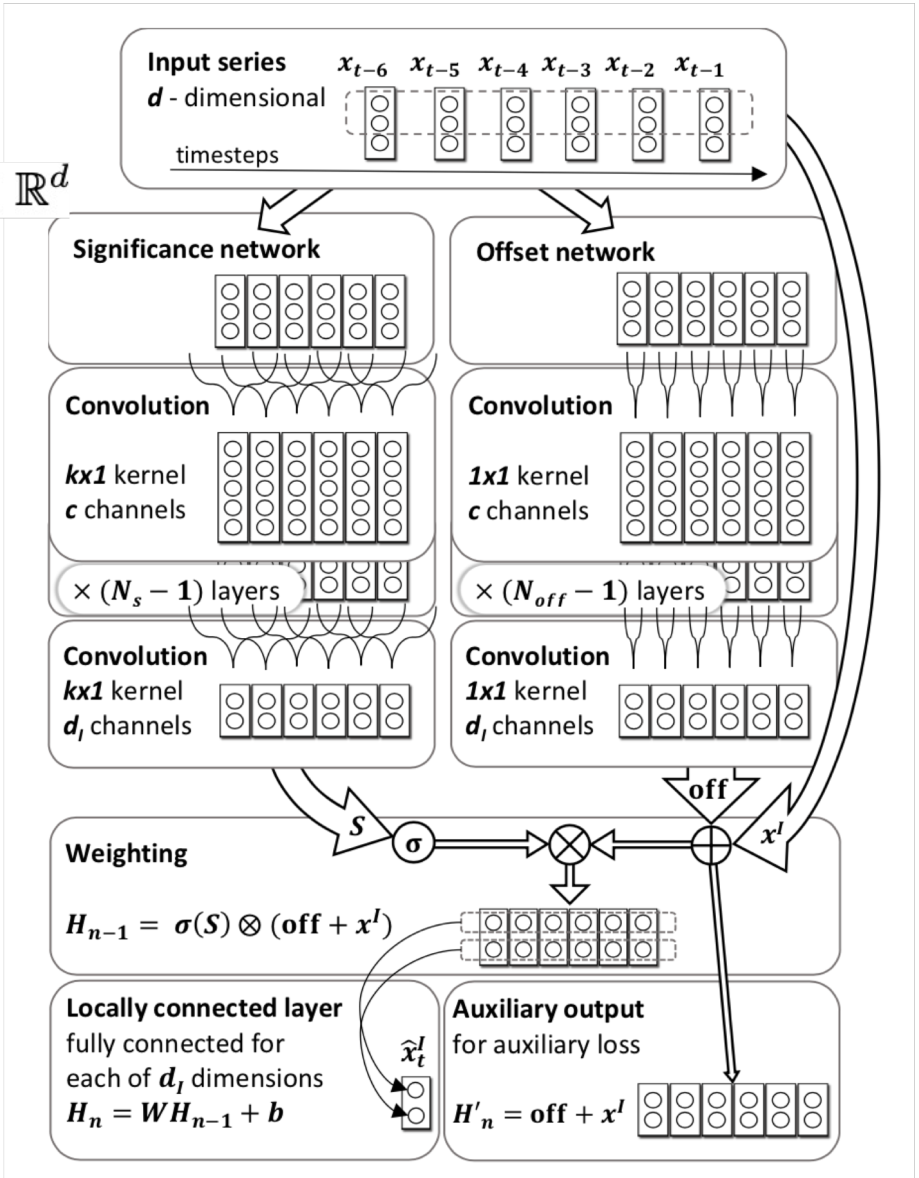
$$y_n = \mathbb{E}[x_n^I | \underbrace{x_{n-m}, m = 1, 2, \ldots]}_{\mathbf{x}_n^{-M}}$$

- The general equation for SOCNN (Significance-Offset Convolutional Neural Network):

$$\hat{y}_n = \sum_{m=1}^{M} [F(\mathbf{x}_n^{-M}) \otimes \sigma(S(\mathbf{x}_n^{-M}))]_{\cdot,m}$$

(similar to Gating and weighting mechanisms mentioned before $f(x) = c(x) \otimes \sigma(x)$)

- F,S are neural networks. S is a fully convolutional network which is composed of convolutional layers only and called significance network.
- σ is a normalized activation function
- ⊗ is element-wise (Hadamard) matrix multiplication.



Model Architecture

- The Eq. of estimator yn can be rewritten as:

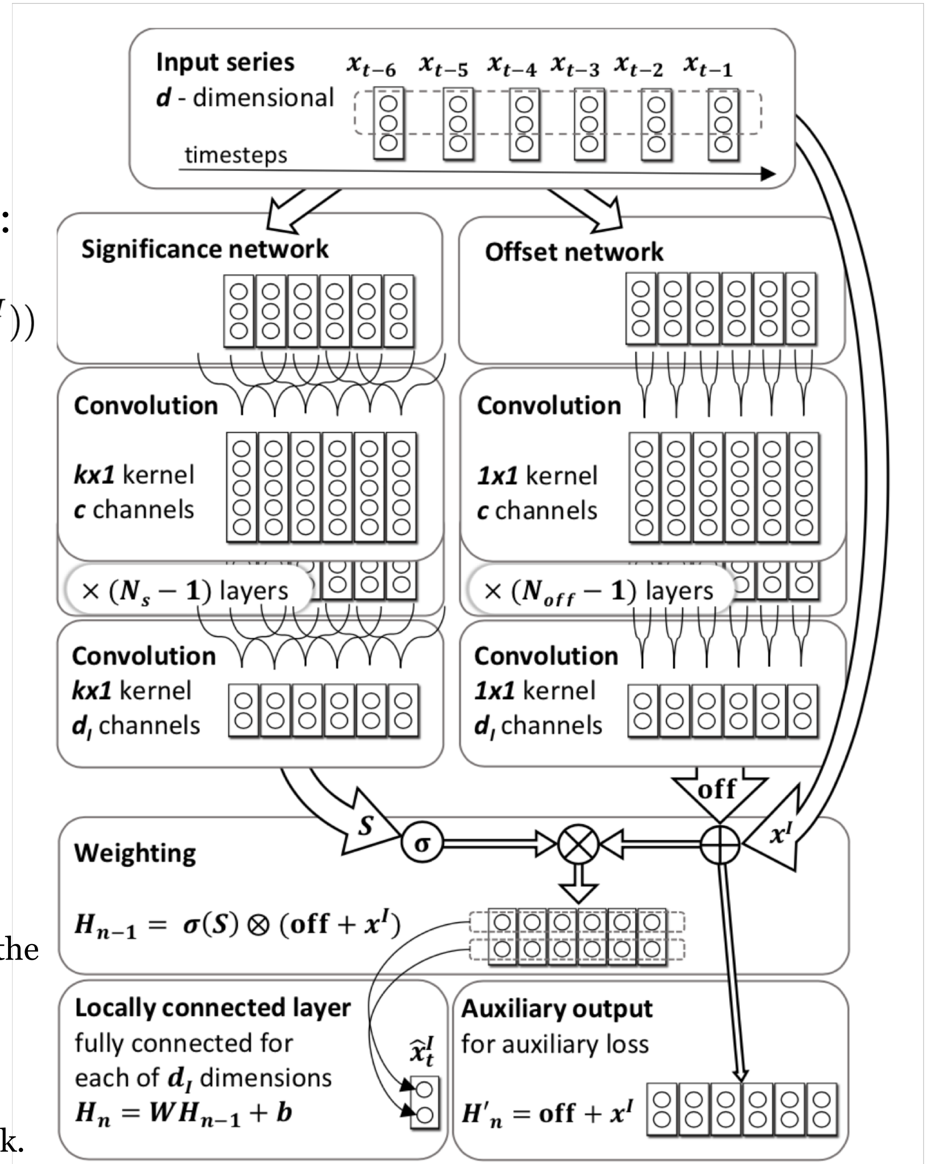
$$\hat{y}_n = \sum_{m=1}^{M} \underbrace{W_{\cdot,m} \otimes (off(x_{n-m}) + x^{I}_{n-m})}_{F(\mathbf{x}_n^{-M})} \otimes \sigma(S_{\cdot,m}(\mathbf{x}_n^{-M}))$$

Significance-Offset Convolutional Neural Network (SOCNN)

- Auxiliary Loss Function:

$$L^{aux}(\mathbf{x}_n^{-M}, y_n) = \frac{1}{M}\sum_{m=1}^{M} ||off(x_{n-m}) + x^{I}_{n-m} - y_n||^2$$

- The eq. enforces the separation of temporal dependence, the local significance of observations Sm, and the predictors off(xn−m) that are completely independent of position in time.
- Each of the past observations provides an adjusted single regressor for the target variable through the offset network.
- Significance network provides data-dependent weights for all regressors and sums them up in an autoregressive manner.



UNIVERSITY OF WATERLOO

Experiments

- Three datasets:
  - Artificially generated datasets: They generated 4 artificial series, $X_{K \times N}$, where $K \in \{16,64\}$. Therefore there is a synchronous and an asynchronous series for each K value.
  - Household electric power consumption dataset: 7 different features excluding date and time.
  - Quotes dataset: 2.1 million quotes from 28 different sources from different market participants, each quote is characterized by 31 features.
- Comparing SOCNN performance with simple CNN, single and multiplayer LSTM and 25-layer ResNet.

UNIVERSITY OF WATERLOO

Training

- Training and validation set: first 80%, sampled by ratio 3:1
  Test set: the remaining 20%
- Adam optimizer
- Batch size=128 for artificial and electricity data
  Batch size =256 for quotes dataset
  batch normalization
- Randomly sampled at the beginning of each epoch
- Dropout and early stopping
- Tensorflow, Keras
- K20s NVIDIA GPU and 8-core Intel Core i7-6700 CPU

Training

- Grid search over some of the hyperparameters
  - depth of offset subnetwork
  - auxiliary weight α
- LeakyReLU activation function, with leak rate a=0.1 :

$$\sigma^{LeakyReLU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ ax & \text{otherwise} \end{cases}$$

  used in all layers except the top layer
- CNN: same number of layers , same stride, similar kernel size, max pooling with pool size=2 every two convolutional layers

Table 1. Configurations of the trained models. *f* - number of convolutional filters/memory cell size in LSTM, *ks* - kernel size, *p* - dropout rate, *clip* - gradient clipping threshold, *conv* - $(k \times 1)$ convolution with kernel size k indicated in the ks column, *conv1* - (1×1) convolution. Apart from the listed layers, each network has a single fully connected layer on the top. Kernel sizes (3, 1) ((1, 3, 1)) denote alternating kernel sizes 3 and 1 (1, 3 and 1) in successive convolutional layers. We also optimized a parameter specific to Phased LSTM, the *initialized period*, considering two: settings: [1, 1000] and [.01, 10].

Artificial & Electricity Datasets					
Model	layers	f	ks	p	clip
SOCNN	10conv + $\{1, 10\}$conv1	$\{8, 16\}$	$\{(3, 1), 3\}$	0	$\{1, .001\}$
CNN	7conv + 3pool	$\{16, 32\}$	$\{(3, 1), 3\}$	$\{0, .5\}$	$\{1, .001\}$
LSTM	$\{1, 2, 3, 4\}$	$\{16, 32\}$	-	$\{0, .5\}$	$\{1, .001\}$
Phased LSTM	1	{16, 32}	-	0	$\{1, .001\}$
ResNet	22conv + 3pool	16	(1, 3, 1)	{0, .5}	$\{1, .001\}$
Quotes Dataset					
Model	layers	f	ks	p	clip
SOCNN	7conv + $\{1, 7\}$conv1	8	$\{(3, 1), 3\}$	.5	.01
CNN	7conv + 3pool	$\{16, 32\}$	$\{(3, 1), 3\}$	.5	.01
LSTM	$\{1, 2, 3\}$	$\{32\}$	-	.5	.0001[10]
Phased LSTM	1	{16, 32}	-	0	.01
ResNet	22conv + 3pool	16	(1, 3, 1)	.5	.01

UNIVERSITY OF WATERLOO

Results

Table 2. Detailed results for all datasets. For each model, we present the average and standard deviation (in parentheses) mean squared error obtained on the out-of-sample test set. The best results for each dataset are marked by bold font. *SOCNN1* (*SOCNN1+*) denote proposed models with one (10 or 7) offset sub-network layers. For quotes dataset the presented values are averaged mean-squared errors from 6 separate prediction tasks, normalized according to the error obtained by VAR model.

model	VAR	CNN	ResNet	LSTM	Phased LSTM	SOCNN1	SOCNN1+
Synchronous 16	0.841 (0.000)	0.154 (0.003)	0.152 (0.001)	**0.151 (0.001)**	0.166 (0.026)	0.152 (0.001)	0.172 (0.001)
Synchronous 64	0.364 (0.000)	0.029 (0.001)	0.029 (0.001)	**0.028 (0.000)**	0.038 (0.004)	0.030 (0.001)	0.032 (0.001)
Asynchronous 16	0.577 (0.000)	0.080 (0.032)	0.059 (0.017)	0.038 (0.008)	1.021 (0.090)	**0.019 (0.003)**	0.026 (0.004)
Asynchronous 64	0.318 (0.000)	0.078 (0.029)	0.087 (0.014)	0.065 (0.020)	0.924 (0.119)	**0.035 (0.006)**	0.044 (0.118)
Electricity	0.729 (0.005)	0.371 (0.005)	0.394 (0.013)	0.461 (0.011)	0.744 (0.015)	**0.163 (0.010)**	0.165 (0.012)
Quotes	1.000 (0.019)	0.897 (0.019)	2.245 (0.179)	0.827 (0.024)	0.945 (0.034)	**0.387 (0.016)**	–

- SOCNN outperforms in asynchronous artificial, electricity and quotes datasets.
- For synchronous data, LSTM slightly better, but SOCNN almost has the same results with LSTM.
- Phased LSTM and ResNet have performed bad on artificial asynchronous dataset and quotes dataset respectively.
- SOCNN1+ has negligible or negative impact.

UNIVERSITY OF WATERLOO

Results

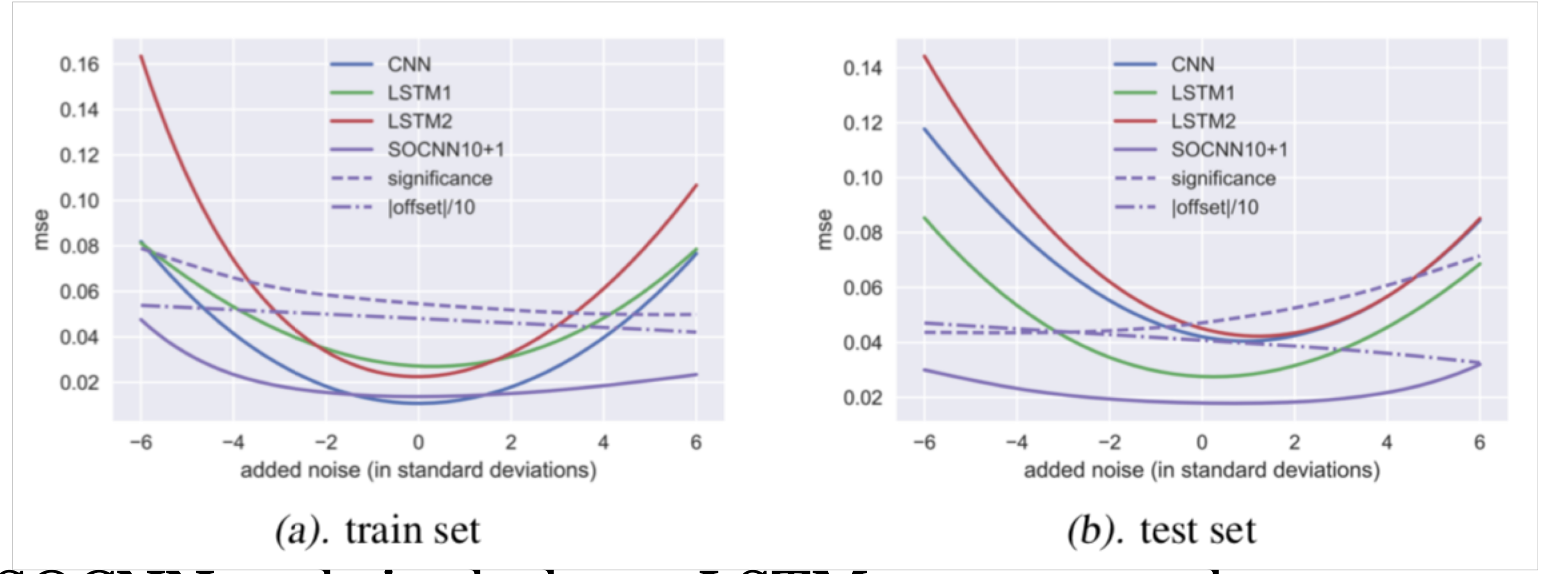
- the higher α improved asynchronous artificial generated dataset, but has negligible impact on other datasets.
- In general, SOCNN had significantly lower variance of the test and validation errors.

Table 3. MSE for different values of α for two artificial datasets.

α	Async 16	async 64
0.0	0.0284	0.0624
0.01	0.0253	0.0434
0.1	0.0172	0.0323

Results

- Test the robustness of the proposed model SOCNN:
  - add noise terms to asynchronous 16 dataset



- SOCNN and single-layer LSTM are most robust compared to other networks.

UNIVERSITY OF WATERLOO

Conclusion

- Significance-Offset Convolutional Neural Network: AR-like weighting mechanism and convolutional neural network.
- High-noise asynchronous time series
- Achieves outperformance in forecasting several asynchronous time series
- Extension:
  - adding intermediate weighting layers
  - not just 1×1 convolutional kernels on the offset sub-network
- econometric datasets

Critique

- The paper is most likely an application paper, and the proposed new architecture shows improved performance over baselines in the asynchronous time series.

BUT

- The quote data cannot be reached, only two datasets available.
- The 'Significance' network was described as critical to the model in paper, but they did not show how the performance of SOCNN with respect to the significance network.
- The transform of the original data to asynchronous data is not clearly stated.
- The experiments on the main application are not reproducible because the data is proprietary.
- The paper didn’t state clearly about the impact and advantage of auxiliary loss function.

Reference

[1] Wilson, A. and Ghahramani, Z. Copula processes. In Advances in Neural Information Processing Systems, pp. 2460–2468, 2010.

[2] Sirignano, J. Extended abstract: Neural networks for limit order books, February 2016.

[3] Borovykh, A., Bohte, S., and Oosterlee, C. W. Condi- tional time series forecasting with convolutional neural networks, March 2017.

[4] Heaton, J. B., Polson, N. G., and Witte, J. H. Deep learn- ing in finance, February 2016.

[5] Neil, D., Pfeiffer, M., and Liu, S.-C. Phased lstm: Acceler- ating recurrent network training for long or event-based sequences. In Advances In Neural Information Process- ing Systems, pp. 3882–3890, 2016.

THANK YOU! ☺