An Augmentation Experiment in Time Series Modeling
(Combination of Input Data Augmentation and 2d Kernels in a CNN)

Nathan Janos and Sangwon Hwang

In the 2019 NeurIPS AutoCV Challenge Kakao Brain took first place using ResNet [1] with an auto-augmentation technique [2]. Neural network models have been a popular area of research for many AI scientists around the world and many neural network models have been studied and developed as a result of these efforts. The techniques developed have proven the robustness and performance capabilities of these methods. In this simple experiment we focus on auto-augmentation for time series data with convolutional neural networks. To test if a CNN can fit and predict time series data with augmentation sub-policies we have chosen extremely high variance data [3] from Criteo Labs.

In the experiment, we chose two simple mathematical methods for augmentation which are the average of a couple of lookback windows and difference equations. In the experiment, we set k to 1, 2 and 4 in the following recurrence relation.

$$\Delta a_n = a_{n+1} - a_n,$$
$$\Delta\Delta a_n = \Delta a_{n+1} - \Delta a_n = a_{n+2} - 2a_{n+1} + a_n,$$
$$\Delta^k a_n = \underbrace{\Delta\Delta\cdots\Delta}_{k} a_n \ \textit{such that } k \geq 1 \textit{ and } k \in \mathbb{N}.$$

Eq 1. Recurrence relation of difference equation

The augmented input data can be represented in a heatmap as can be seen in the following figure. Only the first row is the original data and the other rows are the result of each sub policy.

Fig 1. Heatmap of augmented input data

The CNN is the most popular artificial neural network model in the field of image detection. However, compared to LSTM, the convergence of the hyperparameters of the model is greatly influenced by the characteristics of the current data and it does not take past dynamics into account. It has not been used successfully as a generative model to describe and explain time series data. However, we thought that in this experiment we could better reflect the dependence on the input data using a CNN as the time series model. For more accurate tests we enabled shuffle in PyTorch. The original data and the model predictions can be seen and compared in the figures below.

Fig 2 shows the result of CNN modeling with original time series data. This case tends to have the least MSE in general. From the point of view of evaluation, the first result (Fig 2) has a smaller MSE than the second result (Fig 3) and the second result less than the third result (Fig 5). However, in describing dynamics you can qualitatively judge that the second result is better than the first result since there are strong peaks and other dynamics present. Also, it was found that the combination of augmentation and a 2D kernel results in the model trying to predict the moving trend rather than staying on the baseline which is about 1.8 as seen in Fig 3. Following a moving trend would possibly affect prediction capability. Compared to the well-known ARIMA time series model, the models in Fig 3 and Fig 5 do not show the 'shifting phenomenon' which is shown in Fig 4.

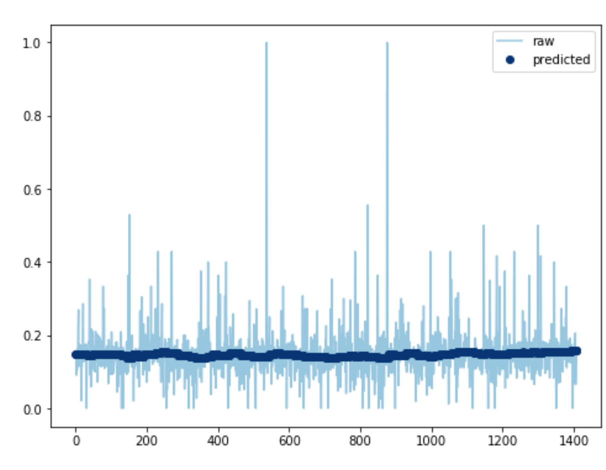

Fig 2. CNN without augmentation

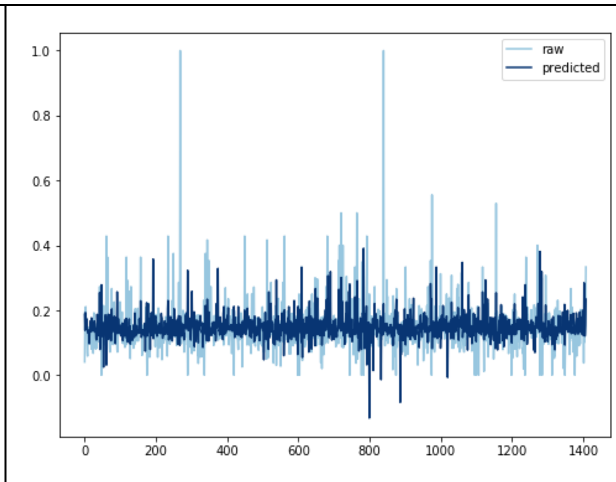

Fig 3. CNN with augmentation and 1D kernel

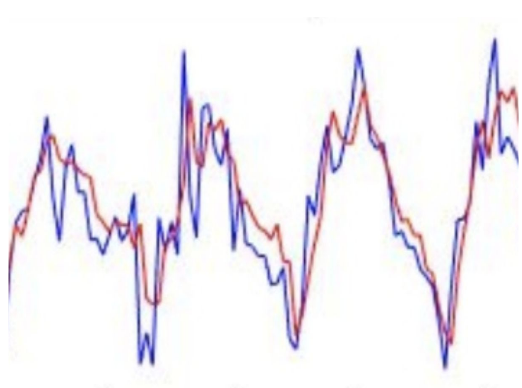

Fig 4. ARIMA

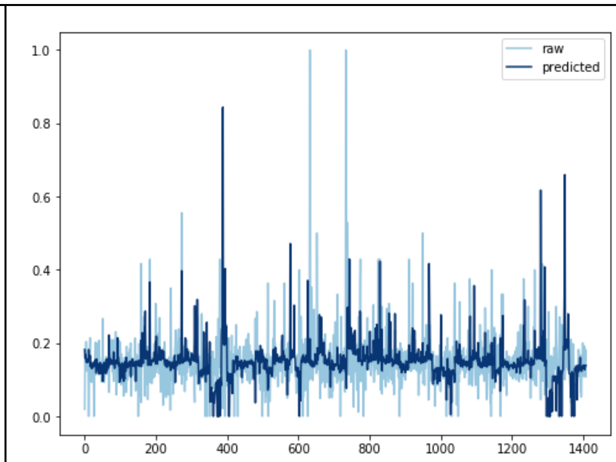

Fig 5. CNN with augmentation and 2D kernel

However, there are definitely limits that we discovered in the experiment. There are many missed peaks in Fig 3 which actually cause higher MSE than Fig 2. Also, the moving trend found with the 2D kernel does not perfectly follow the real trend. In many areas, you can see that the trends do not match.

Next minute CTR prediction is a difficult task due to its extreme variance. With this experiment we verified the possibility and the advantage of using input data augmentation in time series prediction. It is also worth considering other data augmentation functions in addition to the difference function such as parameterized smoothing functions and parameterized aggregators. If we can overcome a few issues found in the experiment this model and method could work well for modeling or predicting time series dynamics. All the code for this experiment is open and available to the public.

References

[1] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016)

[2] Lim, Sungbin, et al. "Fast autoaugment." Advances in Neural Information Processing Systems. 2019.

[3] Chapelle, Olivier. "Modeling delayed feedback in display advertising." Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2014.