Animated Visualization of the Training Process for "Anomaly Transformer: Time Series Anomaly Detection With Association Discrepancy"

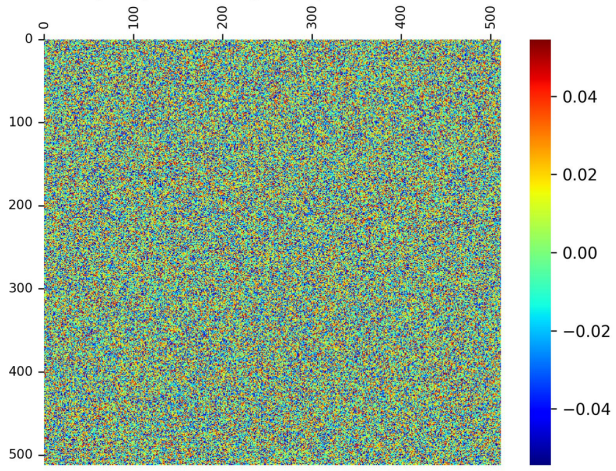
Gilad Erez
Yarden Menashe Eini

July 27, 2023

Git Repository Link

Abstract

In this project, we present animated visualizations of the training process for "Anomaly Transformer: Time Series Anomaly Detection With Association Discrepancy" paper. There are many methods for detecting anomalies, and recently the use of Transformers has entered this field, yielding a significant improvement in performance. The paper introduces a state-of-the-art architecture for anomaly detection in multivariate time series, using a two step "mini-max" optimization. based on the original code, we examined the model's behavior in terms of attention mechanisms and reconstruction abilities over training epochs. We focused mainly on visualising and animating the key and query projection weights, which are the basis of attention in Transformer models. The visualizations aid in understanding the training process, and may serve as an educational resource for transformer-based models in general.



1 Introduction

Our shared interest in time series (in the contexts of medicine and health in which our researches deals), and the contemporary relevance of transformer models, led us to the article "Anomaly Transformer: Time Series Anomaly Detection With Association Discrepancy" by Jiehui Xu, HaixuWu, Jianmin-Wang, and Mingsheng Long.
The paper, which our project is based upon, adapted Transformers (Vaswani et al., 2017) to time series anomaly detection in the unsupervised regime. Transformers have achieved great progress in various areas, including natural language processing, machine vision and time series. This success is attributed to its great power in unified modeling of global representation and long-range relation.

Applying Transformers to time series, the researchers found that the temporal association of each time point can be obtained from the self-attention map, which presents as a distribution of its association weights to all the time points along the temporal dimension. The association distribution of each time point can provide a more informative description for the temporal context, indicating dynamic patterns, such as the period or trend of time series. The above distribution is named "series-association", which can be discovered from the raw series by Transformers.
Time series anomaly detection is a critical aspect of various fields such as finance, healthcare, and cybersecurity.
The outstanding performance of the Anomaly Transformers did not leave a lot of room for improvements, so we decided to explore and visualize the attention mechanisms of the Transformer during the training process.
We decided to visualize this complicated mechanism by plotting and animating the key-query weights, and the reconstruction process in the Anomaly Transformer. Our approach utilizes animated heatmaps to visualize the progression of these weights across different layers, epochs, and batches. This animated visualization approach elucidates how the weights interact and evolve throughout the training, lending a deeper understanding of the model's operational dynamics, and may provide an intuitive understanding of the way the Transformers train and operate.
Additionally, we present a detailed visualization of the reconstruction and reconstruction-error throughout the process. This examination shows the models ability to identify patterns in the series, ignoring anomalies in order to later identify them as unusual.
The data we used for the training is a reduced version of the PSM data-set, which was collected internally from multiple application server nodes at eBay, and has 26 dimensions.

2 Results

In order to show an animated visualization of the attention during the process, we thought about displaying the attention matrices produced in the training process. But, because the attention is a function of the input X, the animation would only show a noisy blur with the rapid change in the input, in a manner that misses the consistent adaptation of the Transformer. For that reason, we chose to go one step backwards and animate the query and key projection weights, which create the attention matrix, and are more robust and independent of the input in the short term.

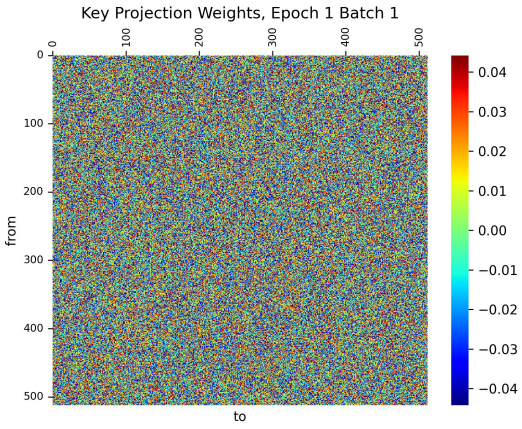


Figure 1: Projection weights heat-map

In figure 1 we can see a heat-map. The title indicates the matrix type (key/query/deltas), and the epoch and batch corresponding to the plot. Each point represents the weight size from the model's dimension (size 512) to the key/query dimension (size 512). The color-bar scale at the right translates colors to numbers, and also indicates the range of numbers present in the matrix itself. All the examples shown here are presented in an animated GIF form in Google-Drive or Github.

2.1 Projection Weights

By examining the 1st layer's key and query projection weights in three different steps in the training process, we can see a few interesting phenomenons.

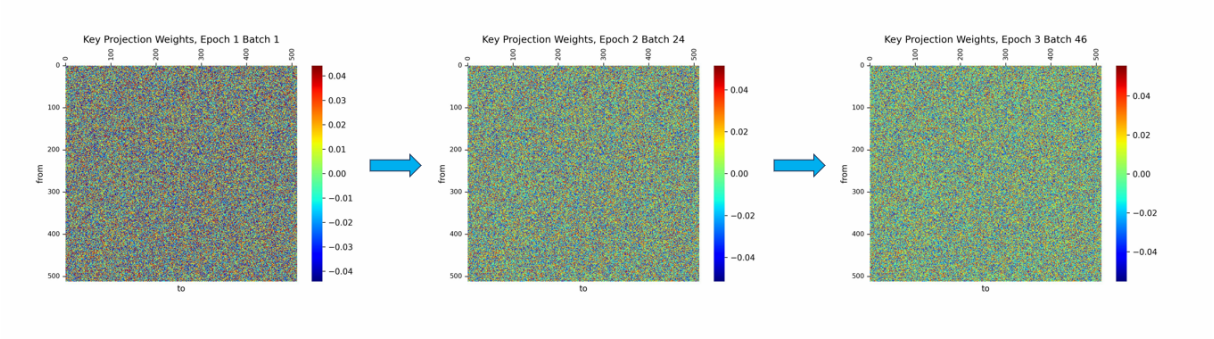


Figure 2: Key projection weights during training process

1. The color-bar scale indicating the range of values in the matrix expands, as at the beginning of the process its limits are at a distance of about 0.04 from zero, and at the end they increase to almost 0.06. This expansion indicates the extremism of certain weights values.
2. The overall heat-map gets greener as the training goes, for two main reasons: The expansion of the color scale that makes medium values relatively closer to 0, and the reduction of medium values weights throughout the process, alongside the extremism of values that were extreme to begin with.
3. The preservation of the patterns that appeared in the weights initialization throughout the training. It can be seen that weights that received extreme values at initialization mostly remained extreme (and even became more extreme), while the other weights decreased in size.

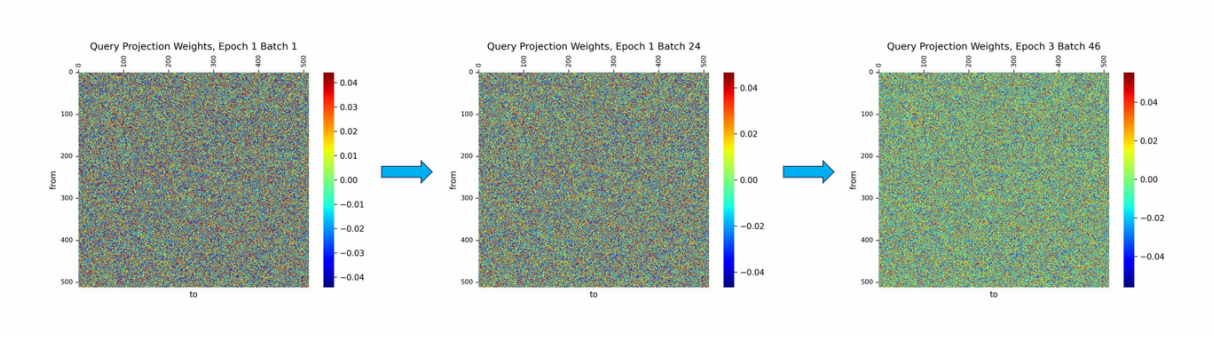


Figure 3: Query projection weights during training process

As much as we can see in these visualizations, query projection weights act similar and present similar conclusions.

2.2 Projection Weights Deltas

Because the changes in the projections heat-maps are not noticeable enough, we decided to also show the "deltas", that is, the changes in weights between one batch and another. With the help of deltas we can compare the behavior of the weights in different parts of the Transformer model.

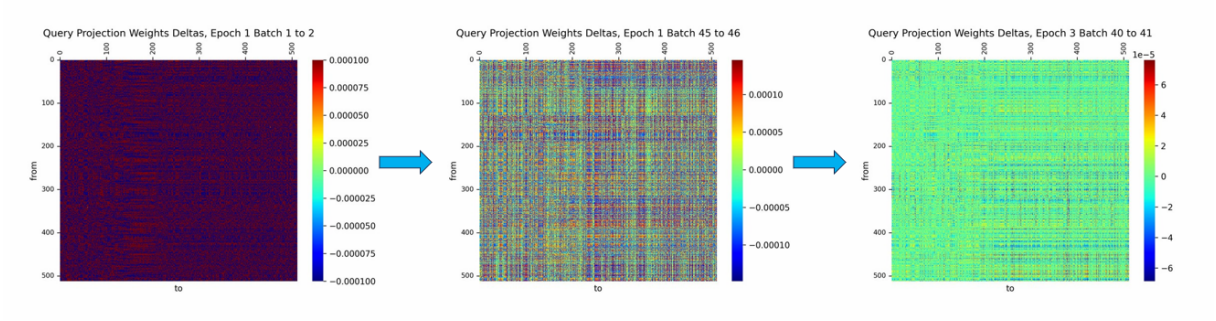


Figure 4: Query projection weights deltas during training process

By examining the 1st layer's key and query weights deltas in three different steps of the training process, we can also see some interesting phenomenons:
1. Unlike in the weights heat-map, the deltas color-bar scale shrinks. This is due to the reducing learning rate along the epochs, and to convergence. There are periods when the scale expends, due to an acceleration of the weights in the beginning of the process, but it shows much better in the animations.
2. The heat-map ends up almost entirely green, with minor changes at the late stages of the training.
3. In this case the patterns seen on the map are horizontal and vertical, and indicate unusual changes in the weights attributed to a feature in the data space (horizontal) or the query space (vertical). We can learn that the importance of certain features is greater than others, and we see this clearly in the graphs.

2.3 Key VS Query Comparison

With the help of deltas we can compare the behavior of the weights in different parts of the Transformer model. We will now analyze the differences between the behavior of the key and the query in the first layer of the model.

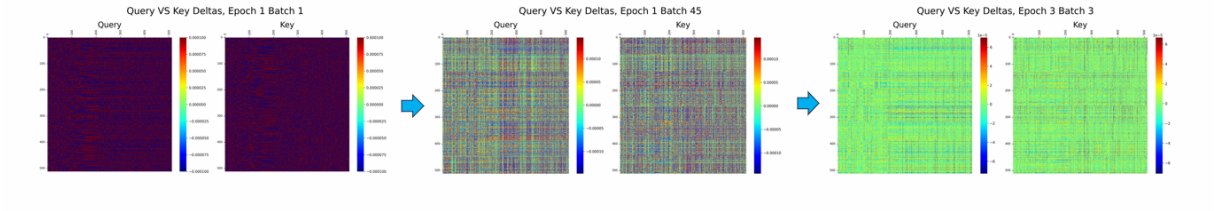


Figure 5: Key and query deltas comparison during training process

It can be seen that the behavior of the weights is very similar over time, that the convergence is timed alike, and that the patterns are similar in both networks. This makes sense, because both networks are subject to the same loss function and affect the operation of the network in a similar way, being projections from one space to another.

2.4 Layer VS Layer Comparison

Another interesting comparison is between the behaviors of the weights in different layers of the Transformer model. The next figure compares the query weights deltas in the 1st and 3rd layers:

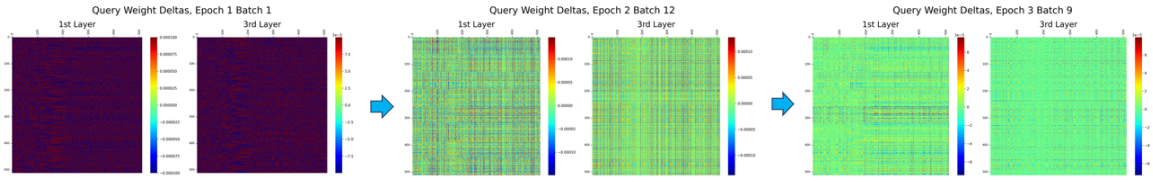


Figure 6: Layer 1 and 3 deltas comparison during training process

In this comparison, the differences between the behavior of the weights in the different layers are evident. These differences are clearer in the animations we attached to the report, but we will describe them here:
1. The convergence rate of the weights is different between the layers. In the early stages, the first layer seems to converge faster, a trend that reverses in the middle of the second epoch. In epoch 3 there are almost no more changes in the weights of the 3rd layer, while in the 1st layer, abrupt and frequent changes in horizontal and vertical patterns are visible.
2. Throughout the training, the 1st layer shows clearer patterns. We assume that the reason is the rawness of the data that passes through it. In the 3rd layer, the raw features are mixed together, and therefore the changes apply to them more equally.

2.5 Reconstruction

We present a plot that visualises the differences between the original signal, shown in blue, and the reconstructed signal, shown in green, across different epochs and batches. At the beginning of the training process, the gap between the signals is quite large. As the training process progresses, it can be seen that the reconstruction of the signal begins to emulate the original signal, thereby reducing the gap between them. This can be seen in the next figure:

Figure 7: Reconstruction and original signal during training process

Another way to present the reduction of the gap as the training process progresses, is a graph which displays the error (difference between the reconstructed and the original signal) in absolute value, at each point in the time window of 100 time points. From each batch we took 5 time series and based on them we calculated the average and the confidence interval (at a confidence level of 95%) of the absolute error. It can be seen in the graph in the early stages of training that the average absolute error is around 1, and as the training progresses the error absolute value reduces to 0.2. Likewise, the confidence interval also diminishes.
It is important to emphasize that the main purpose of the model is to identify anomalies, therefore the loss function is not consist only of an error term. In fact, it is a combination of the a reconstruction-error and an "association discrepancy" measure, which represents the differences between two types of association the model learns, and does not relate to accurate reconstruction.
For that reason, the reconstruction performance of the model are limited.

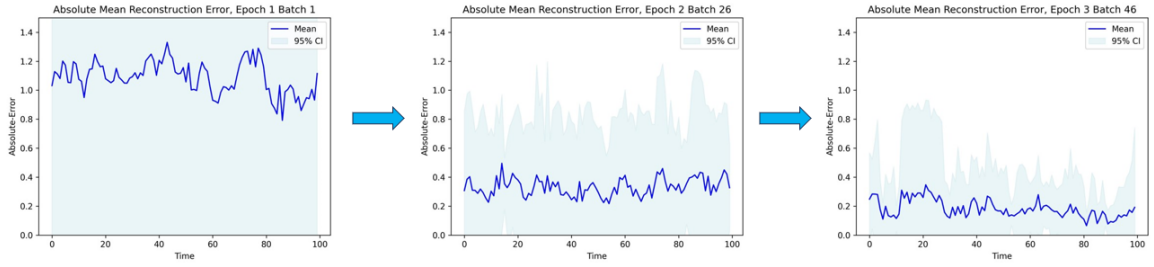


Figure 8: Absolute reconstruction error and confidence interval during the training Process

3 Conclusions

Throughout the project we presented and analysed the learning process of the model in several different aspects.
Looking at the reconstruction capabilities of the model showed an increasingly improving performance, due to an improved modeling of the basic regularity of the time series. However, performance was not perfect due to the secondary importance of this task as part of an anomaly detection task.
Examining the heat-maps of the key and query weights taught us that the final configuration of the networks is very similar in nature to the initialization of the weights. In the learning process, there is an extremism of weights with extreme values, and a moderation of moderate weights, so that the patterns remain. This can indicate the importance of "smart" weights initialization, since the network's ability to wander in the solution space is limited.
The analysis of the key and query deltas taught us about the convergence rate of the weights, and presented the longitudinal and lateral patterns, which are probably created due to the differences in the importance of variables in the data space and the projection space.
We also learned that the key and query weights learning process looks similar, but there is a difference in the process between the different layers, as the later layers converge faster and more uniformly, and with less noticeable patterns.

Your text with citations [XWWL21] and [VSP+17].

References

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 6 2017.

[XWWL21] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. 10 2021.