Two (gyro's) is all you need

Gal Ness*
Physics department, Technion
gn@campus.technion.ac.il

Elad Zohar
Physics department, Technion
elad.zohar@campus.technion.ac.il

Abstract

Wearable technology gyroscopes allow tracking user activities by capturing their angular motion in three axes. We show that, from the information in the temporal evolution of two axes, the third can be predicted by exploiting inter-axis correlations. For this, we employ a multi-head auto-attention transformer adapted to process two input signals. We further show that, using novel positional encoding layers, cross-temporal correlations can boost the prediction accuracy.

1 Introduction

The physical motion of a device, whether a phone or a smartwatch, is usually not restricted to a single axis, and it tends to feature periodic patterns. For example, a running carrying a smartwatch moves his arm in a particular motion, but unless pathologically intending, his movement is not co-linear with the watch's coordinate system. Therefore, the recorded signal would involve accelerations in each of the three gyroscope axes. The primary use of wearable gyroscopes is to set the screen orientation (where is "up"), so slight degradation in accuracy can be bearable for simpler designs and lower end prices. In particular, small futuristic wearables, such as smart rings (1), would strongly rely on miniaturized hardware, so could benefit from simpler elements. Therefore, reducing the gyroscope hardware requirements can significantly contribute a real-life advantage for future wearable designs.

To attain the inter-axis prediction, we use a multi-headed transformer model. Self-attention transformers were designed to handle sequential input data. However, unlike recurrent neural nets, transformers do not process the data in order. Instead, the attention mechanism provides context for any position in the input sequence. A cardinal component of language attention networks is positional encoding, which in our case translates to temporal encoding. We propose nontrivial temporal encoders and comparatively study their performance.

2 Datasets

We use the WISDM (Wireless Sensor Data Mining) dataset for human activity recognition (HAR), which was recorded and labeled in the WISDM lab at Fordham University with the original intention of training activity classification models (2). It contains 18 different labeled activities (labeled A-K) of 51 users, which vary from walking and running to eating pasta and folding clothes, recorded using accelerometers and gyroscopes of both a phone and a smartwatch.

The instances in the dataset are about 3 minutes long, with a sampling rate of 20Hz, which gives about 3600 points of data for each axis (x, y, z). An example from the dataset reads:

```
User, activity,  timestamp,             x,               y,               z
1600,    A,     252207918580802,     -0.85321045,     0.29722595,     0.8901825;
1600,    A,     252207968934806,     -0.8751373,       0.015472412,    0.16223145;
1600,    A,     252208019288809,     -0.72016907,     0.38848877,     -0.28401184;
```

*The project's github repository: **Two-is-all-you-need**

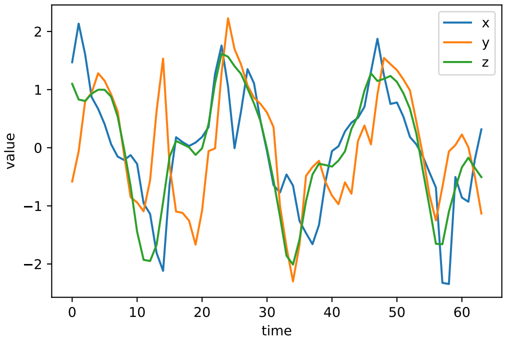


Figure 1: Example of three axes data with temporal correlations. Different colors represent different axes, all normalized to zero mean and unity standard deviation, and plotted vs. time ticks of duration $1/20\text{Hz} = 50\text{ms}$.

We chose to focus our attention on the gyroscope reading of specific watch and phone activities which feature some temporal periodicity, we elaborate on this process in section 4.1.1. The input data therefore includes 300 useful samples overall.

3 Related Work

Previous works that used this dataset focused on activity classification, which was the original goal for acquiring the samples. The classification problem is largely solved (3; 4; 5), but to the best of our knowledge, no attempts were made utilizing this dataset to our use-case. Nevertheless, sequence to sequence signal inference, which is the backbone of this project is a primer problem to solve with transformer models (6). The specific case of using two input sequences to one output sequence and letting the model learn the temporal correlations between the three is more unique. Accordingly, we have not found an existing model to exploit and have implemented a dual transformer of our own.

4 Methods

4.1 Data preprocessing

In this section, we describe the data selection and preparation for usage processes.

4.1.1 Identifying oscillatory data

At first, our model (which is explored in detail in section 4.2) utilized the entirety of the WISDM dataset, and performed rather poorly. Specifically, we noticed the model's output signal would tend to "die out" after a few epochs. This prompted us to first try solving the problem using a more traditional recurrent neural network (RNN). The idea is that if the RNN would also perform poorly, then there is an inherent problem in our data. Using an of-the-shelf RNN, we got similar results to those of the transformer. Our next step was to inspect the dataset a little more carefully and find that a lot of the activities recorded on it showed little to no signal most of the time, and zero periodicity. We thus decided to choose the subset of activities that did exhibit temporally periodic signals, namely, sports activities such as walking, jogging, dribbling etc., this can be justified by the fact that wearable motion sensors are primarily used for recording such activities.

The RNN experiment was thus an important step towards the realization of the transformer.

4.1.2 Sampling and augmentations

The data is divided into 12k sequences, each 80 points (4 seconds) long. An example of such a sequence (sample) is presented in Fig. 1. First, we chop half a sample at the beginning and end of each long recording to avoid irregularities. Then, we reshape the data to have mini-sequences

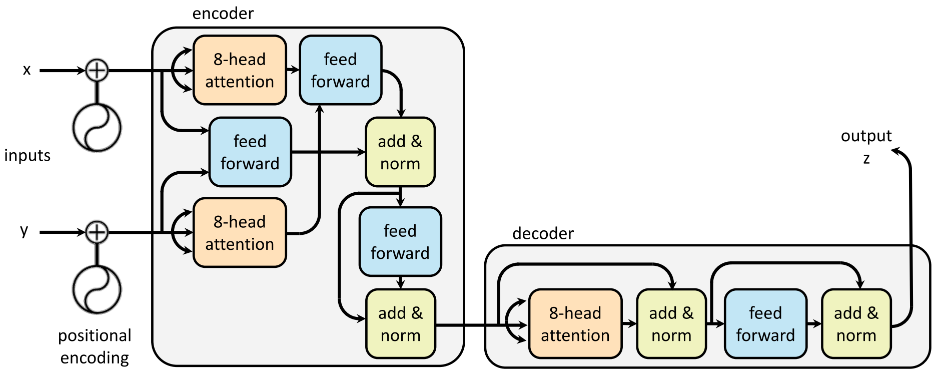


Figure 2: DESAT architecture. A dual encoder design is linked to a single decoder, all containing a self attention mechanism.

of around 10 fold of the desired lengths. The first sample of each mini-sequence is allocated for validation/testing. The rest was chopped with 71 points of buffer on the edges, which allows us to randomly sample the data from a different starting point each epoch, effectively increasing the number of new instances the model "sees" each epoch by a factor of 71. The specific amount of mini-sequences and buffer length automatically vary according to the validation split parameter.

We also use random temporal flipping and sign inversion augmentations to provide even more diversity in the sequences the model encounters. All of which are applied to all three axes simultaneously to preserve the inter-axis correlations.

4.2 Dual encoder self-attention transformer (DESAT)

In this section, we describe the model architectures and the training processes.

4.2.1 Architecture

The DESAT model is a multi-headed attention transformer [see (7)] that utilizes two encoders and one decoder, corresponding to the two axes input to one axis output.

Essentially, we modify the usual sequence-to-sequence transformer by applying two additional linear feed-forward layers: one combining the positional encoded inputs that are fed forward to the decoder without going through the attention mechanism, and another combining the two encoders' attention outputs into a single layer, such that the dimensions of the inputs to the decoder layer are the same as in a vanilla sequence-to-sequence model, but encode data derived from two inputs. Both combining layers mentioned above are fully connected layers, to allow the model to learn the weighted correlation between the three axes. Fig. 2 presents the schematic structure of the model. In addition, the model is designed with integrated positional encoding options, which enable the exploration of a few novel ideas, as detailed in section 4.2.3.

4.2.2 Training Process

The model trains on the training set that is described in section 4.1. The input is a two-dimensional sequence of length 80 points, and the target a one-dimensional sequence of the same length. We chose to use an Adam optimizer initialized with a learning rate of $5 \cdot 10^{-7}$ and a scheduler with a decay factor $\gamma = 0.8$, stepping each 30 epochs. We found that using higher learning rates would lead to the first epochs being completely useless, until the scheduler brings the learning rate to the 10^{-7} regime.

TThe loss criterion we use is the mean-squared-error (MSE) loss, which is most suitable for tasks where the aim is to recover the functional shape of the target at a maximal precision. Explicitly, the

Table 1: DESAT hyperparameters

Epochs	1k
Batch size	128
Learning rate	$5 \cdot 10^{-7}$
Scheduler rate (epochs)	30
Scheduler decay factor γ	0.8
Attention dimension	128
Linear (feed forward) layers dimension	256

MSE of each output is defined as:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \,, \tag{1}$$

where N is the length of the sequence $\mathbf{x}$ (in our case $N = 80$), $\mathbf{y}$ is the target vector, and $\hat{\mathbf{y}}$ is the current model output. All hyper-parameters were manually calibrated and are listed in table 1. The run time per epoch is about 4 second on Nvidia Titan V GPU, so a 1k epochs training takes around one hour.

4.2.3 Positional encoding beyond the Sine

A key ingredient of attention transformers is a positional encoding layer. This is a quasi-spanning set of periodic functions added to the input data to break the translational symmetry of fully connected (attention) layers. The original implementation proposed in (7) used a Sine-Cosine pair of decreasing frequencies to extract the input sequence's positional (temporal) information.

At the point where the model performs sufficiently well (see section 5), we turn to explore different positional encoding concepts which diverge from the mainstream Sine-Cosine set.

The first encoding we treat is the sawtooth function which is attractive for this application for a few reasons. first, it exhibits a temporal behavior similar to that of the sine function, but features a broader Fourier decomposition (see figure 3), which might be exploited to link multiple frequencies in each depth. Second, as the sampling rate of our data is fairly low compared to the frequencies it captures, the data itself shoes sharp-edge peaks, which might resonate with such positional encoding.

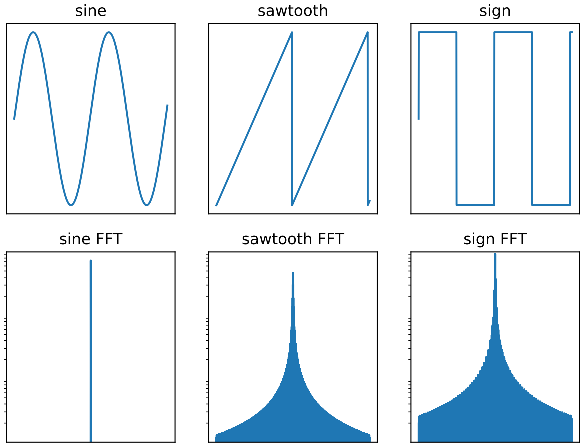


Figure 3: The different functions use for positional encoding and their corresponding Fourier transforms.

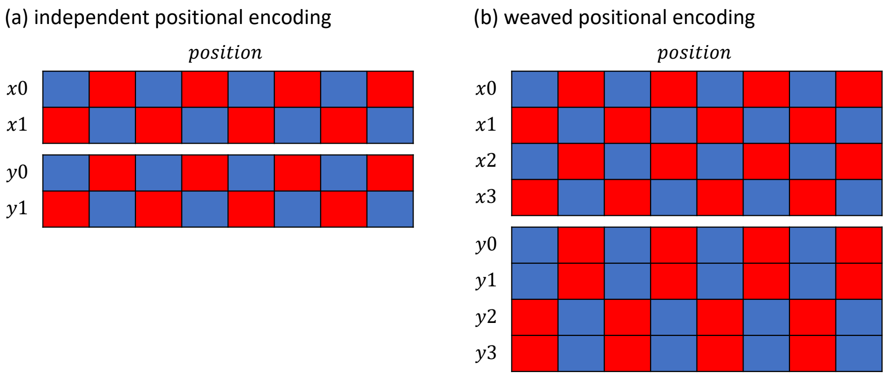

(a) independent positional encoding
position
x0
x1
y0
y1
(b) weaved positional encoding
position
x0
x1
x2
x3
y0
y1
y2
y3

Figure 4: Positional encoding weaving. Example of the first depth layers where the positional encoding frequency kept to its highest. (a) independent positional encoding, both channels (x and y) are used with the same encoding, in this example – sign of sine and cosine. (b) weaved positional encoding, here at each frequency the channels feature one in-phase repetition and one out-of-phase repetition.

The third type of positional encoding we use is a sign function employed on the original Sine-Cosine encoding such that the final encoding is a binary version of the original one. This again gives a very similar temporal (yet more abrupt) behavior, with even broader Fourier spectrum.

A more advanced positional encoding manipulation we explore is weaving. Specifically, we weave the encoding in a more intricately – instead of using the Sine function for even cells and the Cosine for odd cells, we weave another set of Sine + Cosine which are now inlaid every four cells. This method, which is visualized in figure 4 should allow the model to leverage differences in the inter-axis correlations as it embeds an intrinsic phase delay on the positional encoding layer. This again is done for all three (sine, sawtooth, sign) positional encoding types we examine.

5 Results

Figure 5 presents learning curves of models with different positional encoding. Comparing different independent encoding (a) and showing the effect of weaving vs. independent encoding (b).

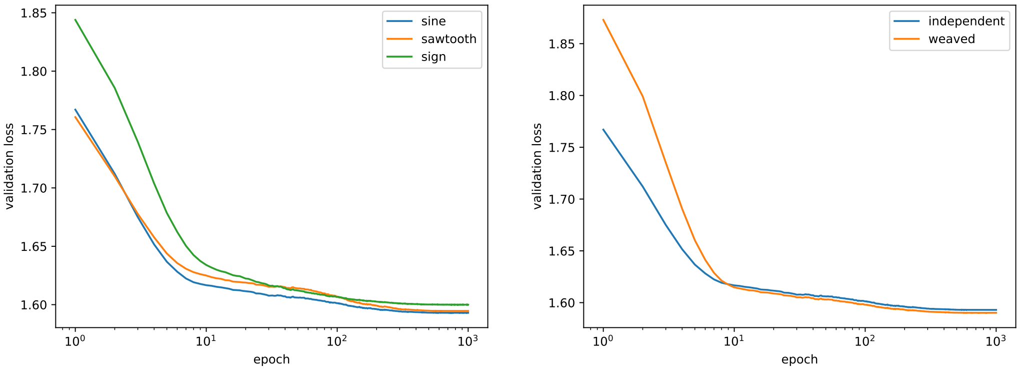
The weaved positional encoding models all showed a similar behavior, where the model started from a worse state but managed to achieve a marginal improvement over long training times with respect to the independent case.

After the model finishes the 1k epochs training, it loops through the validation losses of each epoch and chooses the best one, it then checks the model on the test set and outputs the resulting loss. The best epochs and test losses results are detailed in table 2.

All models discussed above achieved similar results with marginal advantage in favor of the weaved sine model. Moreover, most models achieved optimal performance around epoch 840, which we attribute to the scheduler parameters. Examples of the different model predictions vs. the ground truth are provided in appendix A.

Table 2: DESAT different positional encoding performances

Model	Best epoch	Test loss
Independent sine	839	1.7217
Independent sawtooth	946	1.7215
Independent sign	839	1.7275
Weaved sine	840	1.7178
Weaved sawtooth	946	1.7177
Weaved sign	840	1.7268



(a)

(b)

Figure 5: Validation loss curves for (a) three different independent positional encoding layers, and (b) independent compared with weaved encoding.

6 Discussion and Future Work

In this project, we showed that using an adapted transformer model, inter-axis correlation can make the third axis of a wearable gyroscope redundant, if the data in use exhibits temporal periodicity. This result was then used to explore novel positional encoding schemes. The best performing model we have is in fact the weaved sine model which utilizes the weaving option discussed above. However, future research is required to establish its superiority further.

More generally, we believe this project shows that the concept of positional encoding can be further explored to improve a transformer's ability to perform various learning tasks.

In terms of future work, we see potential in a few directions. First, modern wearables feature 100Hz gyroscopes, which quite possibly could improve the model's ability to perform the task at hand and perhaps even generalize it to other types of data. Second, weaving and other ideas regarding positional encoding are promising research directions that could improve upon the original idea in terms of the learning rate, generalization, and adapting sequential models to noisy data or data that exhibits nontrivial correlations

References

[1] Joseph Nicholi Prencipe. Nfc ring patent, 2021. US Patent 9,313,609 B2. URL: https://patents.google.com/patent/US9313609B2/en.

[2] Gary M. Weiss, Kenichi Yoneda, and Thaier Hayajneh. Smartphone and smartwatch-based biometrics using activities of daily living. *IEEE Access*, 7:133190–133202, 2019. doi:10.1109/access.2019.2940729.

[3] Susana Benavidez and Derek McCreight. A deep learning approach for human activity recognition. *Stanford CS230 course project*, 2019. URL: https://cs230.stanford.edu/projects_fall_2019/reports/26221049.pdf.

[4] Isibor Kennedy Ihianle, Augustine O. Nwajana, Solomon Henry Ebenuwa, Richard I. Otuka, Kayode Owa, and Mobolaji O. Orisatoki. A deep learning approach for human activities recognition from multimodal sensing devices. *IEEE Access*, 8:179028–179038, 2020. doi:10.1109/access.2020.3027979.

[5] Bolu Oluwalade, Sunil Neela, Judy Wawira, Tobiloba Adejumo, and Saptarshi Purkayastha. Human activity recognition using deep learning models on smartphones and smartwatches sensor data. *arXiv preprint*, 2021. URL: https://arxiv.org/abs/2103.03836, arXiv:2103.03836.

[6] Yifei Ding, Minping Jia, Qiuhua Miao, and Yudong Cao. A novel time–frequency transformer based on self–attention mechanism and its application in fault diagnosis of rolling bearings. *Mechanical Systems and Signal Processing*, 168:108616, 2022. doi:10.1016/j.ymssp.2021.108616.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint*, 2017. URL: https://arxiv.org/abs/1706.03762, arXiv:1706.03762.

The project's github repository: **Two-is-all-you-need**

A Prediction examples

Here we collect a couple of examples for each of the six positional encoding DESAT models listed in table 2, comparing the predicted signal (red) and the ground truth (blue).

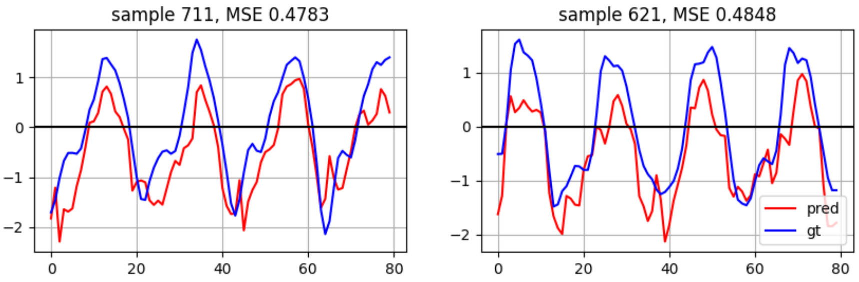


Figure 6: Independent sine positional encoding prediction examples.

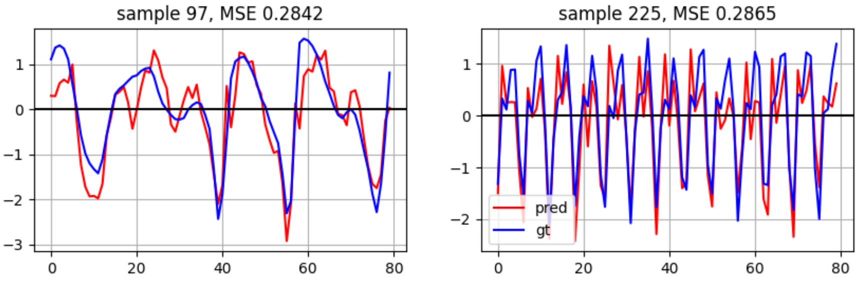


Figure 7: Independent sawtooth positional encoding prediction examples.

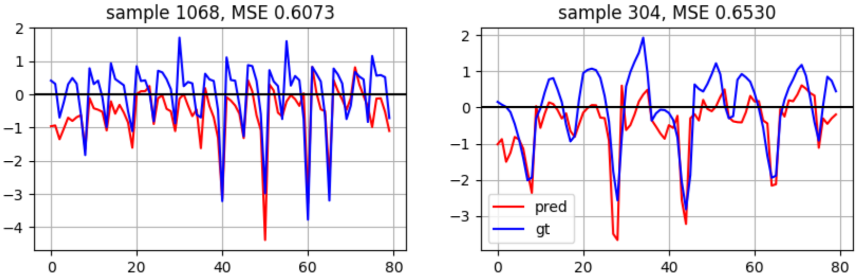


Figure 8: Independent sign positional encoding prediction examples.

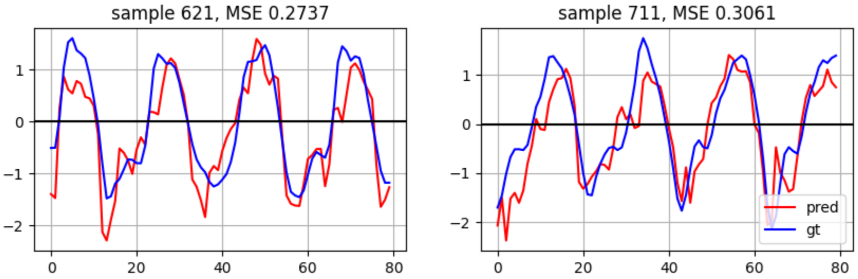


Figure 9: Weaved sine positional encoding prediction examples.

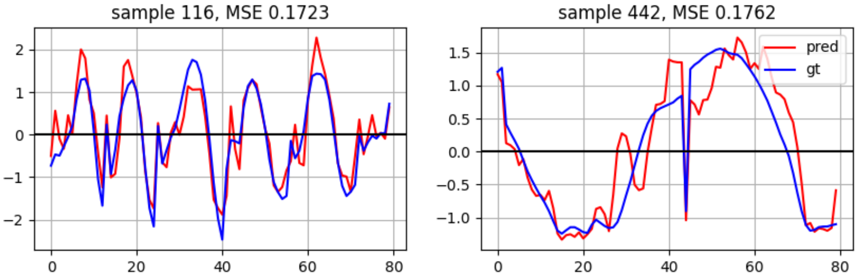


Figure 10: Weaved sawtooth positional encoding prediction examples.

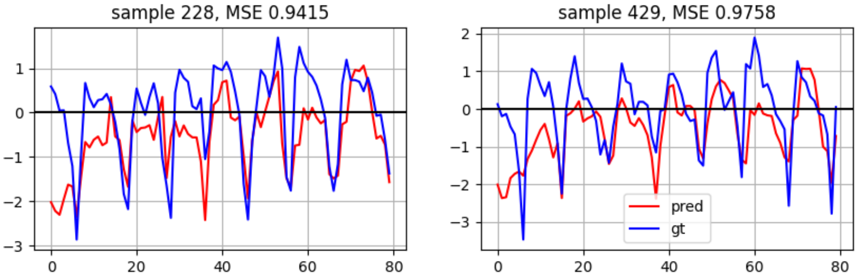


Figure 11: Weaved sign positional encoding prediction examples.