Time2Vec and the Nuances of Temporal Representation

Efstathios Kotsis, Michael Darmanis
Nektarios Christou, Vasileios Venieris

July 22, 2023

Time2Vec, as described in the paper by Borealis et al.[1], offers a vector representation of time, aiming to capture the characteristics of time in machine learning models. However, it can be susceptible to misinterpretation in various domains. This essay tries to explore the potential pitfalls of Time2Vec, especially when juxtaposed against the transformer architecture.

[1] Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. Time2vec: Learning a vector representation of time, 2019

Strengths

1. Time2Vec's ability to encapsulate the progression, periodicity, and scale of time is of great interest. The representation, while being versatile across datasets, captures time's essence in a manner that's more sophisticated than mere timestamps.
2. The transformer architecture, though originally crafted for NLP, has been employed in numerical timeseries classification, as seen in various articles (show articles). Time2Vec's potential as a positional encoding mechanism in such architectures underscores its versatility.

Concerns and Clarifications

1. Many articles seem to use timeseries values (like prices or sales) as inputs to Time2Vec, rather than the actual temporal information (i.e., a time index $t = 0, 1, 2, 3...$). This approach seems more like an enrichment of feature representation rather than true positional encoding. Even the original Word2Vec paper[2], which inspired Time2Vec, hints at its design being more aligned with time-related features.
2. Terms like "positional encoding" have specific meanings. If Time2Vec is used to encode features rather than positions, it leads to potential confusion. Such confusion can compound when architectures are layered and interwoven in complex models.
3. Even if Time2Vec was applied on a time index, it should likely be a global time index created before batching or windowing the data. This is because the local time index within a batch might be arbitrary, potentially introducing inconsistencies from a theoretical point of view.

[2] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013

4. The use of sines in Time2Vec seems to be connected to a Fourier series. Indeed, Fourier series decomposes a function into sines and cosines, potentially capturing periodicities in the data. This raises a crucial question: Is Time2Vec's mechanism analogous to Fourier transformation, and if so, how does it differ in its application?
5. The core idea behind Time2Vec is somewhat related to the Fourier Series in that both deal with the decomposition of signals (or time in the case of Time2Vec) into a set of frequencies. However, while the Fourier Series uses fixed sines and cosines as basis functions, Time2Vec allows these frequencies to be learned.
6. The paper references the use of up to 64 sinusoids. What does using multiple sinusoids imply? It is likely about transforming a single time point into multiple features;[3]
   (a) **Increased Expressivity:** A single sinusoid can only capture one specific frequency or oscillation pattern. By using multiple sinusoids, each with potentially different frequencies and phases, the representation can capture a more complex and diverse set of patterns.
   (b) **Time Decomposition:** Similar to the Fourier transform, which decomposes a signal into its constituent sinusoidal frequencies, using multiple sinusoids allows the representation to decompose the time information into various frequency components.
   (c) **Higher Dimensional Embedding:** By transforming a single time point into multiple features (via multiple sinusoids), the time information is embedded into a higher-dimensional space. This can make it easier for machine learning models to discern patterns, as the higher-dimensional space may separate or disentangle features that are overlapped or conflated in the original time representation.
   (d) **Flexibility:** The ability to learn the parameters of these sinusoids (amplitude, frequency, phase) means the model can adapt the representation to best fit the patterns in the data. For instance, certain frequencies might be more important for one dataset, while others might be more important for a different dataset.
   (e) **Interactions with Other Features:** When a single time point is transformed into multiple features, it provides more avenues for interactions with other features in the data.
7. It employs both linear and sinusoidal transformations. Which features should undergo a linear transformation? Without the original code from the paper, this remains an area of ambiguity and warrants further exploration.

[3] Akin to how one perceives 'now' in terms of day, week, month, and year, such a multi-sinusoidal approach might be an automated way to capture various cyclical patterns in time data.

Conclusion

Time2Vec, in its essence, presents an alternative approach to representing time. By using multiple sinusoids to transform a single time point, it provides a richer, more expressive representation of time, which can capture a diverse range of patterns and make it easier for models to learn from time-based data. However, its application needs careful consideration. Misinterpretations, especially in areas as intricate as positional encoding, can lead to flawed models and misguided conclusions. A more rigorous mathematical framework and clearer guidelines for Time2Vec's application are required.

Acknowledgements

This report was typeset using LaTeX, originally developed by Leslie Lamport and based on Donald Knuth's TeX. A template that can be used to format documents with this look and feel has been released under the permissive Apache License 2.0, and can be found online at `https://tufte-latex.github.io/tufte-latex/`.

References

Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. Time2vec: Learning a vector representation of time, 2019.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.