Time2Vec Embedding on a Seq2Seq Recurrent Network for Pedestrian Trajectory Prediction

Victor Peñaloza[0000−0001−7335−8255]

RLICT: Research Laboratory in Information and Communication Technologies.
Unversidad Galileo, Ciudad de Guatemala, Guatemala.
victorsergio@galileo.edu

Abstract. This paper presents a description of a proposed architecture for the pedestrian trajectory prediction task on TrajNet [16] dataset. The goal of this task is to determine future final agent position and future agent movement trajectory using previous agent positional information. The proposed architecture is a Seq2Seq LSTM recurrent network with a Time2Vec [11] input embedding layer. After experimenting and evaluating, our experimental results show that adding a Time2Vec input-embedding layer improves the performance of a Seq2Seq LSTM recurrent model, on the pedestrian trajectory prediction task.

Keywords: Pedestrian · Trajectory Prediction · Time2Vec · Embedding · Sequence to Sequence · TrajNet.

1 Introduction

Many researchers have studied and discussed the analysis of human mobility due to its applications in robotics, autonomous driving systems, and mobile telecommunications systems. With the growing size of collected data captured by sensors and smartphones at present, it is possible to gather a lot of data about human mobility to be processed later to identify patterns in data [15]. However, for some applications, it is important to have a system capable of anticipating and determining a future human position in an online way, such as the case of autonomous driving systems, in which it is helpful to know in advance the pedestrian movements in streets to avoid accidents.

In this paper, we focused on the prediction of the future path for an agent, using previous movement information to anticipate the future movement. The contributions of this work can be summarized as follows:

1. Show a simple Seq2Seq architecture to predict future pedestrian movement, using an encoder-decoder LSTM recurrent model with an input embedding layer.
2. Compare the performance between a model using an input embedding layer and a model without it. We show experimentally that for pedestrian path prediction on TrajNet [16] dataset, the use of Time2Vec [11] embedding layer improves the Seq2Seq model results.

This paper is comprised of nine sections: the first one presents an introduction to this task and study. The second section describes some previous work in the pedestrian trajectory prediction task. The third section describes the problem statement, metrics, and used dataset. The fourth section presents some informational background of blocks used to build the proposed system. The fifth section describes the system architecture proposed. The sixth section describes the experiments realized. The seventh section shows the results achieved in the model comparisons. The last section presents some conclusions and future work to continue the experiments on this task.

2 Related Work

Many deep learning models have been proposed for human trajectory prediction with data-driven approaches to learning the human motion. LSTM [9] based architectures have been shown that are capable of learning general human movement and predicting their future trajectories [1]. Simple encoder-decoder architectures have been shown good results in pedestrian trajectory prediction, using only position information about pedestrians [4]. Besides, more complex approaches using Generative Adversarial Networks have been used already to predict the motion of pedestrians interacting with others [2].

3 Problem Statement

3.1 Problem Formulation

The goal of this task is to predict a future agent trajectory without using human-human interaction data. We formulate this problem as a sequence generation problem, where we assume agent input trajectories as a sequence $X = (X_1, X_2, \ldots, X_8)$ to predict a future trajectory $X_{pred} = (X_9, X_{10}, \ldots, X_{20})$. Whereas $X_i = (x_i, y_i)$ are (x, y) coordinates in a 2D plane for any time-instant i.

3.2 Dataset Description

For the experiments, we used the TrajNet Benchmark Dataset [16]. This dataset bundle most common datasets used for pedestrian trajectory analysis. TrajNet challenge also provides a common platform to compare path prediction approaches. We train our models using World H-H TRAJ dataset, in which the pedestrian trajectories are formed by x, y world coordinates in meters.

3.3 Metrics and Evaluation

We use the following error metrics to evaluate the performance of the models over a test dataset: average displacement error (ADE) and the final displacement error (FDE). ADE is defined as the average of Euclidian distances between ground

truth and the prediction overall predicted time steps. FDE is defined as the Euclidian distance between the predicted final position and the ground truth final position. We observed eight frames (*2.8 seconds*) to form an input sequence to predict a future sequence comprised of 12 frames (*4.8 seconds*).

4 Background

For a better understanding in this section, we introduced some fundamental concepts of the blocks used to build our model. The following concepts are presented: LSTM [9], Seq2Seq learning, embedding, and Time2Vec [11].

4.1 LSTM, Long Short-Term Memory

Neural Recurrent Networks RNN has been proved an effective architecture for diverse learning problems with sequential data [10]. These networks were designed to deal with vanishing gradients problem in learning long-term dependencies [9].

LSTM architecture mainly is comprised of a memory cell and unit gates. The network can use these gates to decide when to keep or overwrite information in the memory cell, as well as to decide when preventing that other cells will be perturbed by a memory cell output [9].

4.2 Recurrent Sequence-to-Sequence Learning

Seq2Seq is a structure designed to modeling sequential problems in which input sequences and outputs sequences are variable in length [20].

The most common structure associated with Seq2Seq learning is an encoder-decoder structure with recurrent neural networks [18, 3]. Being an LSTM [9] the most used neural recurrent unit in the encoder and decoder section.

The encoder processes an input sequence of m elements and returns representations of z in a fixed-length vector. The decoder takes z and generates an output sequence element at a time [7]. This type of architecture has been used, showing good results in machine translation tasks.

4.3 Embedding

Many factors affect the Machine Learning model's performance; among these factors, the principal factor is the representation and quality of the data used for training [12].

Observing as an example the NLP area in which sequences of sentences, words, or characters are used as input to recurrent neural networks based models; it is observed that generally is desired to provide a compact representation of these features instead of sparse representation [19]. Diverse techniques have been proposed for feature representation in NLP as one-hot encoding representation [21], continuous bag of words(CBOW) [14], and word-level embedding [8].

Word level embedding goal is to learn a representation in which ideally words semantically related are highly correlated in the representation space [19], in this way good results have been obtained in classification and sentiment analysis NLP tasks.

4.4 Time2Vec Embedding

In many applications with sequential data, time is an important feature. In many models using recurrent neural networks, the time is not taken in count as a feature. When time is observed as an important factor, generally, this is added as an additional feature [5, 6, 13].

Time2Vec is a vector representation (*embedding*) to time that it can be combined easily with other architectures [11].

Time2Vec can represent three important features of time, periodically, time scaling invariance, and simplicity to be added to existent features [11].

For a given scalar notion of time τ, Time2Vec of τ, denoted as $\mathbf{t2v}(\tau)$, is a vector of size $k+1$, defined as follows:

$$\mathbf{t2v}(\tau)[i] \begin{cases} \omega_i \tau + \varphi_i, & \text{if } i = 0 \\ \mathcal{F}(\omega_i \tau + \varphi_i), & \text{if } 1 \leq i \leq k. \end{cases}$$

Where $\mathbf{t2v}(\tau)[i]$ is the i^{th} element of $\mathbf{t2v}(\tau)$, $\mathcal{F}$ is a periodic activation function, and ω_is and φ_is are learnable parameters.

When using $\mathcal{F} = sin$, for $1 \leq i \leq k$, ω_i and φ_i are the frequency and the phase-shift of the sine function.

This function permits learn periodic behaviors. The linear term section can be used to represent no periodic behaviors [11].

5 System description

The main idea of the proposed system is the combination of already existing techniques that have shown improvements in other research areas as NLP, and apply these techniques to the pedestrian trajectory prediction task.

We proposed the use of a recurrent Seq2Seq architecture that uses bidirectional [17] LSTM layers to build the encoder and decoder with a Time2Vec embedding layer on encoder inputs to feed a better positional sequence representation to the model. Fig. 1 shows the proposed architecture for pedestrian trajectory prediction.

We choose a Seq2Seq multi-step architecture because it has been observed that single-step prediction models that use windowing to predict a complete future path are prone to error accumulation, and Seq2Seq models have been showing better results in similar tasks [20].

Although Time2Vec originally was proposed for a better time representation, we can see the sequences of coordinates (*positions in the space*) as signals that include an implicit time relation, and use Time2Vec embedding to create an embedding that represents this signal with a learnable frequency and phase shifts of *sin* function.

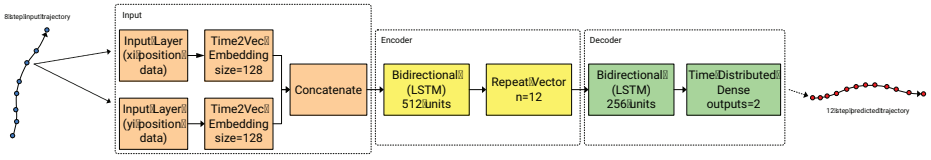

Fig. 1. Proposed Seq2Seq + Time2Vec embedding architecture.

6 Experiments

For performance comparison purposes, we build a Seq2Seq LSTM architecture that not uses Time2Vec embedding and compares it against the proposed Seq2Seq LSTM model with Time2Vec embedding, to observe if the use of a Time2Vec embedding is beneficial to the performance of Seq2Seq model in trajectory prediction results.

We used 20% of training data as validation, the Seq2Seq model without embedding was trained for 87 epochs, and the Seq2Seq + Time2Vec model was trained for 94 epochs. Both models were trained using a learning rate = 1e-5, batch size = 8, and MSE loss function. The embedding size for the Seq2Seq + Time2Vec model was set to 128.

For comparison, we used error metrics average displacement error (ADE) and the final displacement error (FDE) in meters.

Furthermore, we run an additional experiment using an embedding vector size of 256 to compare the effect of embedding size in the model's performance.

7 Results

Table 1. Comparison of prediction error among different models to evaluate embedding performance.

Model	Embedding size	ADE	FDE
Seq2Seq + Time2Vec	128	0.573	**1.559**
Seq2Seq + Time2Vec	256	**0.565**	1.62
Seq2Seq	-	0.657	1.782

Table 1 shows our experimental results, it can be observed that the use of Time2Vec embedding helps the Seq2Seq model to predict trajectories that are more accurate. Time2Vec embedding helps the model feeding it with a better representation of positional trajectories sequences.

Notice that the run with a bigger embedding vector size shows a downgrade in the model's performance when final displacement error (FDE) is used as a metric, but even the results are better than the model without the embedding layer.

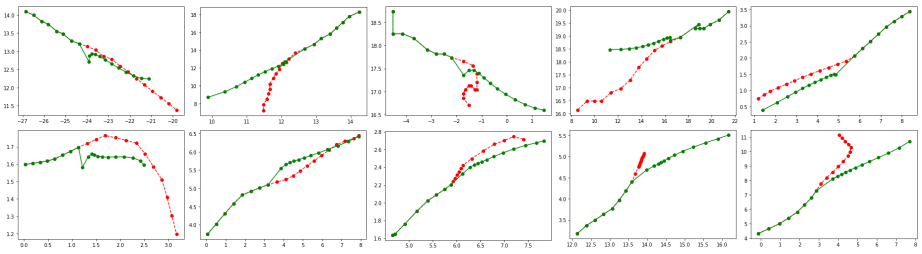

Fig. 2. Some examples of predicted trajectories. The ground truth trajectories are shown in red and model's predictions are shown in green. The scale of the graph axis is in meters.

8 Conclusions and Future Work

In this work, we described an architecture that uses a Seq2Seq LSTM recurrent model with an input embedding layer to be compared with a similar architecture without an input embedding layer, to show the benefits of having an adequate input data representation for models on the pedestrian trajectory prediction task.

According to our experimental results, although the use of embedding improves the performance of the proposed architecture in this task, now the embedding vector size hyper parameter appears. It can be observed that the size of this vector has a model's performance impact and must be adjusted correctly.

Additionally, it can be worthy of testing other architectures as GAN based, transformed based and CNN based for comparing embedding performance in the pedestrian trajectory prediction task.

Acknowledgments

This work was supported by Facultad de Ingeniería de Sistemas, Informática y Ciencias de la Computación (FISICC) and Research Laboratory in Information and Communication Technologies (RLICT), both part of Universidad Galileo from Guatemala.

We thank Jean-Bernard Hayet, PhD. for helpful advice and research directions related to this work.

References

1. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social lstm: Human trajectory prediction in crowded spaces. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 961–971 (2016)
2. Amirian, J., Hayet, J.B., Pettré, J.: Social ways: Learning multi-modal distributions of pedestrian trajectories with gans. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) pp. 2964–2972 (2019)

3. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. CoRR **abs/1409.0473** (2015)
4. Becker, S., Hug, R., Hübner, W., Arens, M.: Red: A simple but effective baseline predictor for the trajnet benchmark. In: ECCV Workshops (2018)
5. Choi, E., Bahadori, M.T., Schuetz, A., Stewart, W.F., Sun, J.: Doctor ai: Predicting clinical events via recurrent neural networks. Proceedings of machine learning research **56**, 301–318 (2016), https://app.dimensions.ai/details/publication/pub.1084501054
6. Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., Song, L.: Recurrent marked temporal point processes: Embedding event history to vector. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 1555–1564. KDD '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2939672.2939875, https://doi.org/10.1145/2939672.2939875
7. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.: Convolutional sequence to sequence learning. In: ICML (2017)
8. Goldberg, Y.: Neural network methods for natural language processing. Synthesis Lectures on Human Language Technologies **10**(1), 1–309 (2017). https://doi.org/10.2200/S00762ED1V01Y201703HLT037, https://doi.org/10.2200/S00762ED1V01Y201703HLT037
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**, 1735–1780 (1997)
10. Karpathy, A., Johnson, J.E., Fei-Fei, L.: Visualizing and understanding recurrent networks. ArXiv **abs/1506.02078** (2015)
11. Kazemi, S., Goel, R., Eghbali, S., Ramanan, J., Sahota, J., Thakur, S., Wu, S., Smyth, C., Poupart, P., Brubaker, M.A.: Time2vec: Learning a vector representation of time. ArXiv **abs/1907.05321** (2019)
12. Kotsiantis, S.B., Kanellopoulos, D., Pintelas, P.E.: Data preprocessing for supervised leaning. World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering **1**, 4104–4109 (2007)
13. Li, Y., Du, N., Bengio, S.: Time-dependent representation for neural event sequence prediction (2018), https://openreview.net/pdf?id=HyrT5Hkvf
14. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013)
15. Morzy, M.: Mining frequent trajectories of moving objects for location prediction. In: Perner, P. (ed.) Machine Learning and Data Mining in Pattern Recognition. pp. 667–680. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
16. Sadeghian, A., Kosaraju, V., Gupta, A., Savarese, S., Alahi, A.: Trajnet: Towards a benchmark for human trajectory prediction. arXiv preprint (2018)
17. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing **45**(11), 2673–2681 (1997)
18. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. ArXiv **abs/1409.3215** (2014)
19. Torfi, A., Shirvani, R.A., Keneshloo, Y., Tavaf, N., Fox, E.A.: Natural language processing advancements by deep learning: A survey (2020)
20. Wang, C., Ma, L., Li, R., Durrani, T.S., Zhang, H.: Exploring trajectory prediction through machine learning methods. IEEE Access **7**, 101441–101452 (2019)
21. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Proceedings of the 28th International Conference on Neural In-

formation Processing Systems - Volume 1. p. 649–657. NIPS'15, MIT Press, Cambridge, MA, USA (2015)