

Traffic Modelling Through a LSTM Variational Auto Encoder Approach: Preliminary Results

Stefano Chiesa and Sergio Taraglio(✉)

ENEA, Robotics Laboratory, ENEA, Rome, Italy
{stefano.chiesa,sergio.taraglio}@enea.it

Abstract. The work in progress on a LSTM sequence to sequence variational autoencoder generative model is presented. The architecture is trained on a floating car data dataset in order to grasp the statistical features of the traffic demand in the city of Rome. An analysis of parameters influence is furnished. The generated trajectories are briefly compared with the ones in the dataset. Further work direction is provided.

Keywords: Generative models · Variational auto encoder · Urban mobility · Traffic model

1 Introduction

Transportation forecasting is the attempt of estimating the number of vehicles or people that will use a specific transportation facility in the future. This can be considered as a relevant subject of Urban Computing: the process of tackling the major issues which cities face using big and heterogeneous data collected by a diversity of sources in urban areas [1].

Among the different approaches to the problem of traffic modelling, activity-based models are a group that try to predict for individuals where and when specific activities (e.g. work, leisure, shopping) are carried out [2]. The idea behind activity-based models is that travel demand is derived from activities that people need or wish to perform; travel is then seen as just a sort of by-product of an agenda, as a component of an activity scheduling decision. In this work we are mainly interested in modelling the traffic flows in an urban environment.

Traffic modelling begins with the collection of data. People can be of extreme help in this, since they generate data while travelling e.g. by carrying and using a mobile phone. Data can be collected through many different means, e.g. using CDR (Call Data Records) from the cellular phones, social network posting, vehicle or smartphone navigation data, floating car data (FCD) et cetera. The approach here presented is loosely related to activity-based models since it exploits car trajectory stops in a floating car dataset (FCD) which can be linked to human activities such as working, sleeping, shopping. However the FCD can account for a very partial description of human activities, only the car related ones, but they can help in reconstructing some features of mobility using machine learning.

© Springer Nature Switzerland AG 2021
O. Gervasi et al. (Eds.): ICCSA 2021, LNCS 12950, pp. 598–606, 2021.
https://doi.org/10.1007/978-3-030-86960-1_43

Human mobility can be seen as a highly structured domain composed of mostly regular daily/weekly schedules, showing high predictability across diverse populations [3]. Further analysis also suggests that human mobility shows temporal and spatial regularities [4, 5].

All these mobility data suffer from a major drawback: the issue of anonymization, the travelling information should be gathered namelessly in order to protect the people's right to privacy. Besides this basic fact, there are further concerns linked to the actual availability of data, since they often belong to private enterprises or governmental agencies, and linked to their representativeness, being often biased or with a low sampling rate.

A possible way to overcome these concerns is to employ generative models, as it has recently been proposed in several fields of application [6], and also to tackle the modelling of traffic demand [7, 8].

In a nutshell a generative model can be described as follows. Let us assume a dataset of observations with a given unknown statistical distribution P_{data}. A generative model P_{mod} is designed to mimic P_{data}, thus in order to generate new observations distributed as P_{data} it would be possible to sample P_{mod} [6]. The model is trained with the observation data and it can learn a compact feature space, whose sampling outputs new observation data with the same statistical properties of the original ones. This clearly overcomes on one hand the problem of user privacy and on the other the risk of too small datasets.

A reliable approach for the generation of new patterns on the basis of the knowledge of a given domain is represented by the use of auto encoders. A further refinement of these is represented by the variational auto encoders (VAE) [6], which are able to add a stochastic component that helps in producing both good data modelling and generation at the same time.

In the following a machine learning approach and some preliminary results are presented on a FCD dataset composed of the trajectories of a set of private cars in the town of Rome, in order to reproduce the traffic flows and vehicle geographical density. The architecture used is a sequence to sequence Long Short Term Memory variational auto encoder, the LSTM being a recursive neural network proficient in modelling sequences [9].

In the second section of the paper a brief description of the chosen architecture is provided together with a sketch of standard and variational auto encoders and LSTM neural networks. In the third the dataset is described and some data processing disclosing some interesting features together with the data representation is presented. In the fourth the training results and some preliminary analysis on generated data are provided. Finally in the fifth section some conclusions and future work directions are given.

2 LSTM Networks and the Variational Auto Encoders

Recurrent neural networks (RNNs) have become the state of the art for sequence modelling and generation, the language and translation domains being the more instantiated subjects. Long Short Term Memory (LSTM) networks [9] are one of the most popular variations of RNNs with proven ability to generate sequences in various domains, such as text [10], images [11] and handwriting [10]. The success of RNNs in these domains,

which present similarities with the traffic one, has motivated several works applying RNNs for human activity sequence modelling and generation [7, 8, 12].

An auto encoder is an artificial neural network used to learn efficient data encodings in an unsupervised manner. The auto encoder is generally composed of two subsections: an encoder and a decoder, they are linked by an internal latent space. The trick is that the latent space dimension is significantly smaller than the input one, which is the same as output, so the model is forced to discover and efficiently internalize the essence of the data. During the training, the system learns to reproduce in output the same data given in input. Once that it has been trained, it is possible to generate new output data via the sampling of the latent space. A further refinement of these models is represented by the variational auto encoders which are able to add a stochastic component in the latent space that helps in producing a better data modelling and generation, see [13].

In this paper, a variational auto encoder based on a sequence-to-sequence LSTM architecture has been used. A LSTM encoder encodes a sequence of places visited by a driver during a temporal interval of one day to its stochastic latent space and a symmetrical LSTM decodes it to the output trajectory sequence, which is made as matching as possible to the input one during the training phase. Once trained the latent space can be normally sampled in order to produce trajectory sequences similar, in a statistical sense, to the ones of the training dataset.

3 The Dataset

The available Octo-Telematics dataset is a FCD containing one month of records from a set of about 150K cars. Each record consists of time, position, speed, distance from preceding record point, engine status (running, turned off, started) and numerical ID of the car. The records are stored as soon as either a given time interval has passed or a given distance is travelled; also the starting or switching off of the engine are recorded events. In short, the dataset is composed of geographical points along car routes.

The first step has been that of studying the dataset in order to extract some global characteristics. As an example in Fig. 1 is shown an analysis in terms of stops, where a stop has been defined as a turning off of car engine for more than 5 min. The average number of stops per day on the whole dataset is 4.99 and 4.04, respectively in weekdays or weekends/holidays. It is remarkably similar to the results in [7] hinting to a sort of universality of human travel behaviour.

In Fig. 1, the bivariate histogram of stops in terms of the stop starting time and the stop duration is plotted. As it can be seen, three areas are clearly recognizable: very short-term stops, overnight stops and the 9-to-5 working stops.

From this and similar analyses it has been chosen to use a simplified dataset composed of sequences containing a maximum number of eight stops in a day. In the following, we will refer to these sequences as trajectories. Each stop is described by six quantities: longitude, latitude, duration, start time, day of week and trip distance covered to reach the current stop. Thus a trajectory is represented by 48 values; in case the number of daily stops is smaller than eight, the data are padded with zeroes; if a given driver performs more than eight stops, it is discarded (all the trajectories of the driver are discarded, to avoid data distortion).

With the above assumption a fraction of 50% of all the available drivers are considered.

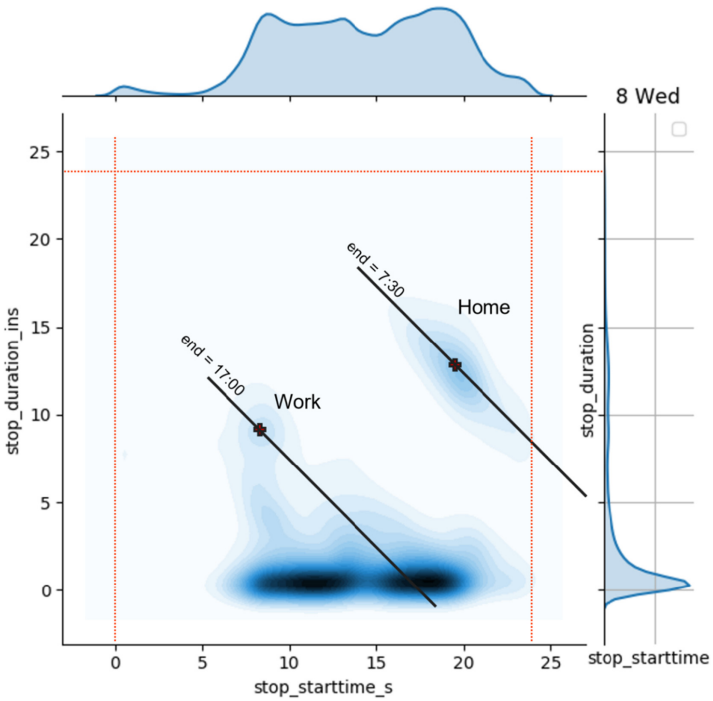


Fig. 1. Histogram of the stops on Wed. 8 May 2013. In x the starting time of the stops and in y the relative duration, both in hours. The red dotted lines delimit the day, the intensity of color increases with the number of stops. The two diagonal lines represent the set of stops ending respectively at 17:00 and 7:30 next day, hinting at the workplace stops and nightly residential ones

4 Training and Generation

4.1 Training

An LSTM sequence-to-sequence VAE has been trained on the subset of the dataset comprising the drivers with the considered number of stops, that means more than 900 K different trajectories. An analysis on the influence on training of the various features of the architecture has been performed and is here presented.

The loss function of a VAE is usually composed of two terms: a reconstruction loss describing how well the output matches the input and a regularisation term helping in furnishing a multivariate normal distribution in the latent space. In this work the two terms are tuned through the:

$$L_{TOT} = \beta L_{KL} + (1 - \beta) L_{MSE} \tag{1}$$

where L_{MSE} is the reconstruction loss, typically measured by the mean squared error, and L_{KL} is the regularisation loss, typically assessed with the Kullback-Leibler divergence. In Fig. 2 is shown the final loss behaviour versus the β term. It is evident that it is sufficient a very low regularisation term contribution to obtain good results.

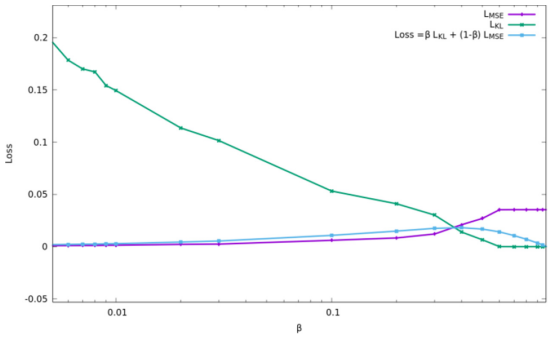


Fig. 2. Final loss behaviour as a function of β the trade-off parameter between reconstruction error (MSE) and regularisation term (KL), a value for β of 0.02÷0.03 assures a low MSE with good regularisation

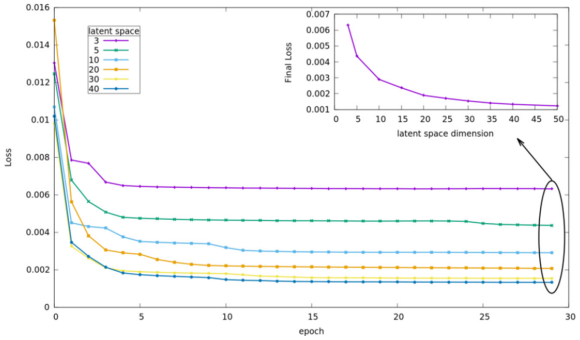


Fig. 3. Training with different latent space dimensions. Larger graph: the total loss as a function of training epochs for different latent space dimensions. Inset: the final loss behaviour as a function of latent space dimension ($\beta = 0.02$)

Several latent space dimensions have been tested and in Fig. 3 is summarized both the training loss as a function of epochs and the final one. It is evident that the larger the latent space dimension the smaller the loss. In the following, for further analyses, the latent space dimension has been chosen at 20.

In Fig. 4 is shown the loss function behaviour versus the LSTM size, i.e. the number of LSTM neurons in both the encoder and decoder parts of the auto encoder. The final loss becomes rapidly insensitive to the number of LSTM cells in the autoencoder, thus a number of 32 LSTM neurons has been chosen.

In Fig. 5 is shown, instead, the final loss behaviour as a function of the number of different drivers considered, i.e. 0.1 K, 1 K, 3 K, 10 K and 30 K. It is noteworthy that

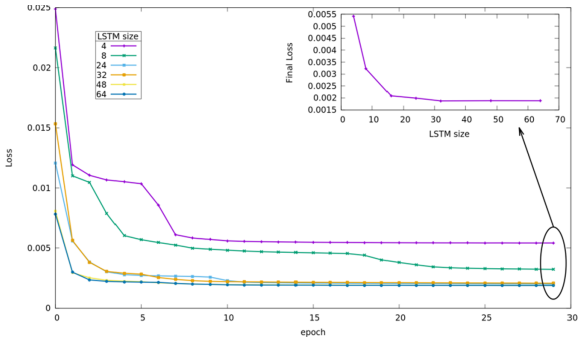


Fig. 4. Training with different LSTM numerosity. Larger graph: the total loss as a function of training epochs for different LSTM sizes. Inset: the final loss behaviour as a function of LSTM size ($\beta = 0.02$)

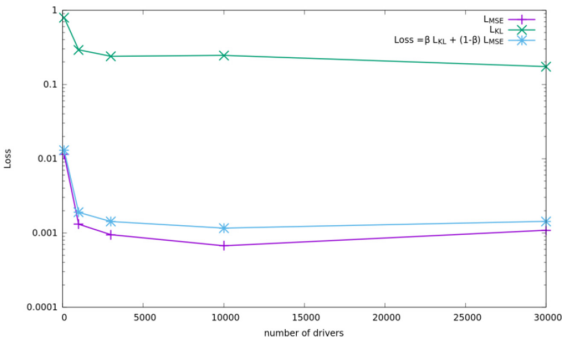


Fig. 5. Final loss behaviour as a function of the number of drivers, with $\beta = 0.02$, and a latent space dimension of 50

the error remains roughly constant while the dataset increases in number in both the re-construction and the regularisation terms, the plot is relative to runs with 100 training epochs, a latent dimensionality of 20, 32 LSTM neurons and $\beta = 0.02$.

The overall reconstruction error is in the order of 0.1%, that can be translated into a geographical position error in the order of a kilometer.

4.2 Generation and Results

Once trained, the system has been used for its true purpose: the generation of new trajectories for synthetic cars and drivers. The here reported data are representative of the generation of 30 K trajectories.

In Fig. 6 the normalized distribution of the number of daily stops in both the training dataset and the generated trajectories are presented. The comparison shows that the statistical distribution of this quantity has been remarkably grasped by the generative model.

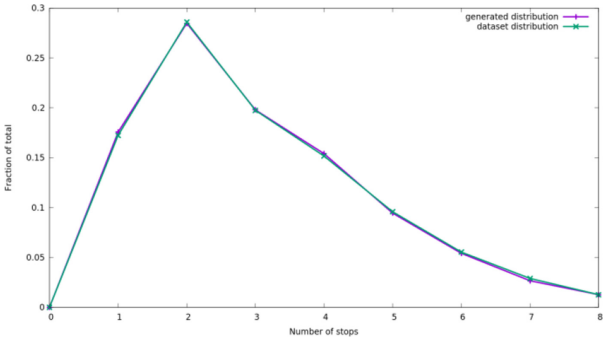


Fig. 6. The normalized distribution of the number of stops in the trajectories: comparison between the one from the training dataset and the one from the generated trajectories

In Fig. 7 the same bivariate histogram of Fig. 1 is presented, i.e. stop start time versus stop duration.

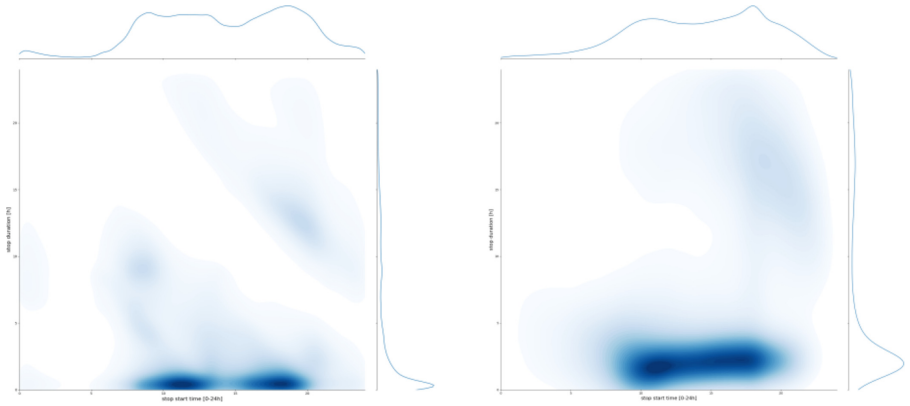


Fig. 7. Histogram of the stop start time vs duration: left the training dataset, right the generated trajectories

On the left of the figure there is the original training dataset, while on the right the generated trajectories. The short-term stops are in the same time slots and with slightly longer durations, the nightly stops in the generated trajectories are there but somewhat longer in duration and later in their beginning, nonetheless there is an evident similarity between the two graphs and the overall quality of the generated trajectories is similar to the original dataset behaviour.

The overall results are quite encouraging hinting to the ability of the generative system to grasp the general statistical distribution of the trajectories in the training dataset.

5 Conclusions and Future Work

It has been presented a generative model based upon a LSTM sequence to sequence variational auto encoder used to model and generate human motorized mobility pat-terns deduced from a FCD dataset.

The here presented preliminary results show that the proposed approach is able to generate trajectories with overall statistical features similar to the ones of the training set. Some tuning of the model has been performed in terms of latent space dimensionality, encoder/decoder size and loss function trade-off between the reconstruction term and the regularization one.

Future work is directed towards two different paths: on one side the experimentation of architectures different from the LSTM to build the encoder and decoder will be considered, specifically both a CNN (convolutional neural network) and a vanilla dense network are going to be tested. On the other side a tiling approach for the geographic localization will be tailored, subdividing the spatial domain into small tiles, e.g. hexagons with a diameter of 700 m, since the goal of the work is the modelling of traffic needs in terms of origin-destination matrices. The validation of the foreseen implementations will be performed also against different and larger traffic datasets.

Acknowledgements. This work has been partly carried out in the framework of the Triennial Plan 2019–2021 of the National Research on the Electrical System (Piano Triennale 2019–2021 della Ricerca di sistema elettrico nazionale), funded by the Italian Ministry of Ecologic Transition.

References

1. Zheng, Y., Capra, L., Wolfson, O., Yang, H.: Urban computing: concepts, methodologies, and applications. ACM Trans. Intell. Syst. Technol. (TIST) **5**(3), 38 (2014)
2. Chu, Z., Cheng, L., Chen, H.: A review of activity-based travel demand modeling. In: Proceedings of the Twelfth COTA International Conference of Transportation Professionals, pp. 48–59 (2012). https://doi.org/10.1061/9780784412442.006
3. Gonzalez, M.C., Hidalgo, C.A., Barabasi, A.L.: Understanding individual human mobility patterns. Nature **453**(7196), 779–782 (2008)
4. Song, C., Qu, Z., Blumm, N., Barabasi, A.L.: Limits of predictability in human mobility. Science **327**(5968), 1018–1021 (2010)
5. Alessandretti, L., Sapiezynski, P., Lehmann, S., Baronchelli, A.: Evidence for a conserved quantity in human mobility. Nat. Hum. Behav. **2**, 485–491 (2018). https://doi.org/10.1038/s41562-018-0364-x
6. Harshvardhan, G.M., Gourisaria, M.K., Pandey, M., Rautaray, S.S.: A comprehensive survey and analysis of generative models in machine learning. Comput. Sci. Rev. **38**, 100285, ISSN 1574–0137 (2020). https://doi.org/10.1016/j.cosrev.2020.100285
7. Yin, M., Sheehan, M., Feygin, S., Paiement, J., Pozdnoukhov, A.: A generative model of urban activities from cellular data. IEEE Trans. Intell. Transp. Syst. **19**(6), 1682–1696 (2018). https://doi.org/10.1109/TITS.2017.2695438
8. Lin, Z., Yin, M., Feygin, S., Transportation, M.S., Paiement, J., Cee, A.P.: Deep generative models of urban mobility. In: Proceedings of KDD 2017, 13–17 August 2017, Halifax, Nova Scotia, Canada (2017)

9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
10. Graves, A.: Generating sequences with recurrent neural networks. arXiv:1308.0850 (2013)
11. Gregor, K., Danihelka, I., Graves, A., Rezende, D.J., Wierstra, D.: Draw: a recurrent neural network for image generation. arXiv:1502.04623 (2015)
12. Huang, D. et al.: A variational autoencoder based generative model of urban human mobility. In: IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), pp. 425–430, San Jose, CA, USA (2019). https://doi.org/10.1109/MIPR.2019.00086
13. Kingma, D.P., Welling, M.: An introduction to variational autoencoders. Found. Trends Mach. Learn. **12**, 307–392 (2019)