Recurrent Conditional GANs for Time Series Sensor Modelling

Edvin Listo Zec [*1] Henrik Arnelid [*2] Nasser Mohammadiha [2 3]

Abstract

Simulation of the real world is a widely researched topic in many different fields, and the automotive industry in particular is very dependent on real world simulations. These simulations are needed in order to prove the safety of advance driver assistance systems (ADAS) and autonomous driving (AD). In this paper we propose a deep learning based model for generating time series outputs from sensors used in autonomous vehicles. We implement a Recurrent Conditional Generative Adversarial Network (RC-GAN) consisting of Recurrent Neural Networks that use LSTMs in both the generator and the discriminator in order to generate sensor errors described as time series that exhibit long-term temporal correlations. The network is trained in a sequence-to-sequence fashion where we condition the model output with time series describing the environment, which enables the model to capture spatial and temporal dependencies. The RC-GAN is used to generate time series describing the errors in a production sensor on a data set collected from real roads, and yields significantly better results as compared to previous works on sensor modelling.

1. Introduction

A lot of progress is continuously being made in the race to autonomy in the automotive industry. Many advance driver assistance system (ADAS) technologies, such as lane keeping assist, collision warning and emergency braking are already implemented in vehicles today and are foreseen to reduce the risk of accidents on roads. In order to make a safe decision, the vehicles are installed with a lot of different sensors such as cameras, radars and lidars.

In order to guarantee the safety of the autonomous vehicles with a certain confidence, billions of miles have to be driven in order to provide a good statistic for a low fatality rate (Kalra & Paddock, 2016). This is a very time consuming and expensive task which cannot be done in reality. Companies producing software for ADAS and autonomous driving (AD) thus resort to virtual verification, where models of the environment and sensors are made to match reality. Modelling a sensor can be done in many different ways on different levels. For example, it is possible to model the raw detections of a sensor. In this paper we focus on object level data, which is the output of sensor fusion. Modelling sensor characteristics such as sensor errors is a challenging problem because it requires models that can capture stochastic behaviours of sensors. Furthermore, it is important to understand how the errors correlate with different traffic scenarios and settings.

Generative models are a fundamental part in a lot of different machine learning algorithms and the attention to generative models is increasing, a lot due to their capability of modelling underlying statistical structures of high dimensional signals. Especially in computer vision, Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) have had great success recently generating realistic high-quality images. Inspired from this, in this paper we develop a deep learning based sensor model capable of modelling the real-valued time series data describing errors in sensor outputs using adversarial networks.

2. Related work

The work in this paper is related to previous works on sensor modelling and also to Generative Adversarial Networks and Recurrent Neural Networks for time series generation and prediction. Since the introduction of GANs, they have shown good results in generating realistic samples in many different applications (Isola et al., 2017; Ledig et al., 2017; Reed et al., 2016), where tasks involving images are predominant. However, the amount of research for generating continuous real-valued time series using GANs is limited as

[*]Equal contribution [1]RISE Research Institutes of Sweden, Gothenburg, Sweden [2]Zenuity, Gothenburg, Sweden [3]Chalmers University of Technology, Gothenburg, Sweden. Correspondence to: Edvin Listo Zec <edvin.listo.zec@ri.se>, Henrik Arnelid <henrik.arnelid@zenuity.com>.

Time Series Workshop at International Conference on Machine Learning, Long Beach, California, 2019. Copyright 2019 by the author(s).

compared to image generation. GANs have previously been used for sequential data generation, but these typically focus on discrete outputs such as in language processing (Yu et al., 2017). In (Mogren, 2016) the author uses an RNN based GAN in order to generate classical music in the form of continuous sequences. In (Esteban et al., 2017) the authors develop a similar RNN based GAN to generate continuous medical time series. In (Donahue et al., 2018) they modify already existing image generation methods to operate on audio waveforms, which is a different approach than using GANs for modelling multivariate time series.

On the topic of sensor modelling, an Autoregressive Input-Output Hidden Markov Model (AIO-HMM) for generation of real-valued time series describing sensor errors has recently been proposed in (Listo Zec et al., 2018). Given input features describing the environment, the authors generated time series similar to that exhibited by a sensor. The AIO-HMM is an extension to the standard HMM, where the next output is conditioned on the previous output and where both the outputs and the hidden states are conditioned on an input vector. However, there is no internal memory in the AIO-HMM and it has some trouble modelling long sequences. Given the recent success of GANs, we implement a modified version of the RC-GAN as described in (Esteban et al., 2017) and show that it is possible to improve the realism of the generated time series for using an internal memory.

3. Problem Overview

The output from sensors used in ADAS and AD considered in this paper is in the form of dynamic state vectors over time, describing variables such as object position, velocity and acceleration relative to the host vehicle collecting the sensor data. These outputs from production sensors inherently exhibit noise and inaccuracies. The main contribution in this paper is developing a model for generating synthetic but realistic production sensor outputs, conditioned on the environment. Particularly, we focus on modelling the time series describing the longitudinal and lateral position and velocity errors from the sensors. A trained model can then be used in Computer Aided Engineering (CAE) tools for virtual testing.

In the same fashion as in (Listo Zec et al., 2018), we use a radar and camera fusion based production sensor setup in our experiments. Moreover, the ego vehicle is also equipped with a Velodyne lidar HDL-64E which is used as a reference system. The lidar data is processed using object classification and tracking algorithms and the output is in the form of object lists with estimated properties like position and speed. We define the production sensor error as the difference between the production sensor and the reference sensor outputs for every detected object over time.

In order to calculate the error we need to associate each production sensor object with a corresponding reference system object. We do this by using an offline matching algorithm (Florbäck et al., 2016), where output from the algorithm is a matrix consisting of object properties in the form of time series for each object. The data set that we use consists of sensor outputs collected from drives on European highways, spanning over 12,000 multivariate time series. The test set contains around 2,000 multivariate time series. The average length of all time series is 197 frames with a minimum of 50 and maximum of 828 frames. In this paper we focus on results regarding the error of longitudinal positions.

4. Model

4.1. Generative Adversarial Networks

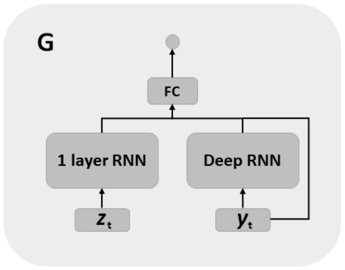
A Generative Adversarial Network (GAN) is a generative neural network that aims to generate samples given a distribution $p_{\text{data}}(\boldsymbol{x})$ of the training data. In the GAN architecture there are two different neural networks trained simultaneously, a generator $G(\boldsymbol{z};\boldsymbol{\theta}_g)$ and a discriminator $D(\boldsymbol{x};\boldsymbol{\theta}_d)$, which have conflicting objectives. The generator learns a distribution p_g over the data $\boldsymbol{x}$, whereas the goal of the discriminator is to discriminate between the synthetic data $G(\boldsymbol{z})$ generated by G and the real data $\boldsymbol{x}$. In practice, this is a minimax game problem described as $\min_G \max_D \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$ with $p_{\boldsymbol{z}}(\boldsymbol{z})$ as a prior over the input noise variables.

4.2. Recurrent Conditional Generative Adversarial Networks

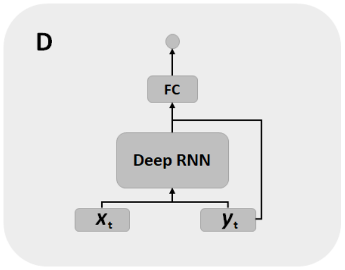
The model implemented in this paper is based on the networks described in (Esteban et al., 2017). The first difference from the original GAN that the authors propose in their paper is that both the generator and discriminator are replaced by RNNs with LSTM units. The second change is that the output from both the generator and the discriminator is conditioned on an input vector $\boldsymbol{y}$, which makes it possible for the GAN to learn the conditional probability distribution $p(\boldsymbol{x}|\boldsymbol{y})$ as described in (Goodfellow et al., 2014). The minimax problem is then described as $\min_G \max_D \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x}|\boldsymbol{y})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z}|\boldsymbol{y})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})|\boldsymbol{y}))]$.

In this paper, we do not generate labels as done in (Esteban et al., 2017). In our case, G and D are conditioned on multivariate time series which are predictive of the response variable, e.g. angle, heading and speed. D takes as input either a real or synthetic time series, and outputs a softmax probability at each time step classifying if the sample is real or not. Figure 1 depicts the network architecture used in this paper.

One main change in our model as compared to (Esteban et al., 2017) is that we isolate the latent noise to its own RNN in G. The second main change is that skip connections are added, connecting the input condition directly to a fully connected layer before the output neuron(s). This allows the generator to predict from both memory and current information. The generator thus consists of two sets of RNNs, one that the latent vector $\boldsymbol{z}_t$ is fed to and one that the conditional input is fed to. The output from both RNNs and the skip-connected condition data are concatenated and fed into a fully connected layer and the final output is a linear activation. By isolating $\boldsymbol{z}_t$ it is possible to control the amount of noise that is applied to the output in much greater detail through varying the distribution that $\boldsymbol{z}_t$ is sampled from, the network size and the size of $\boldsymbol{z}_t$.



(a) Generator network (G). It takes input from a latent space as well as condition data $\boldsymbol{y}_t$ at each time frame.



(b) Discriminator network (D). It takes either a real or synthetic time series together with the condition data $\boldsymbol{y}_t$ as input at each time frame.

Figure 1. The architecture of the generator (top) and discriminator (bottom).

The discriminator consists of a structure where data point(s) from either a real or synthetic time series $\boldsymbol{x}_t$ is being concatenated with the corresponding condition data $\boldsymbol{y}_t$ at each time frame and is fed to an RNN-LSTM network. A skip connection is also used in this network. By adding the skip connection for $\boldsymbol{y}_t$ to the fully connected layer for both networks, a more stable and faster learning was obtained during training. The best model in this paper consists of two-layered RNNs in both the generator and the discriminator.

5. Model evaluation

Evaluation of generative models in general and GANs in particular is an open research question that is far from solved (Theis et al., 2015). Current evaluation still relies on human validation to asses the quality of the generated sample. Further, in our case we seldom have more than one realisation from an underlying unknown stochastic sensor model. Using only one sample makes it difficult to draw any reasonable conclusions of the model and its quality.

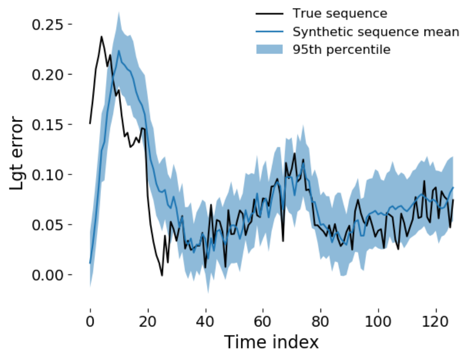
We evaluate our model by using a test set including around 2,000 multivariate time series of lengths between 50 and 828. For each error sequence in the test set, we input the corresponding multivariate time series into the model and generate 100 synthetic sequences. In order to evaluate if different parts of the signals, such as large errors, are represented with the right density in the generated sequences, we use the Jensen-Shannon distance (JSd) (Briët & Harremoës, 2009). Further, we also use the root mean squared error (RMSE) to assess temporal dependencies. In addition to this, we also use the JSd between the first difference distributions of the test set and the generated samples.

6. Results and Discussion

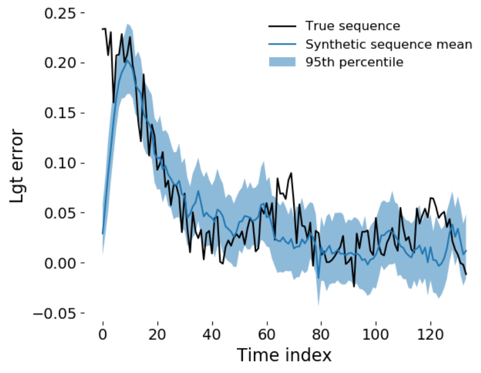
We randomly initialised and trained several different RC-GANs and evaluated each model using JSd and RMSE. In Figure 2, three different test sequences together with corresponding generated sequences from the best performing model are visualised. These sequences were chosen to illustrate different behaviour in the data and how the model performs in these cases. The test sequence of the longitudinal position error is plotted with a black line, while the mean of 100 generated sequences from the RC-GAN is plotted with a blue line. The filled blue area is the 95th percentile of all generated sequences. All trained models got a substantial initial transient for all generated time series. We see that other than the initial transient, the generated sequences behave similar to the real sequences. This transient makes the RC-GAN behave poorly in generating the first time frames as seen in Figure 2. When the network has been fed enough time frames, it starts to be able to grasp the context of the sequences and the output starts to look more real.

We also trained the generator network G using the mean absolute error (MAE) as loss function instead of the regular GAN training setting, i.e. without the discriminator. These two models and the AIO-HMM are compared in Table 1. In Figure 3, the distribution of generated time series for each test sequence is shown in orange and the distribution of the test sequences is shown in blue. Figure 3 and Table 1 show that the distributions are very similar with a JSd of 0.082.

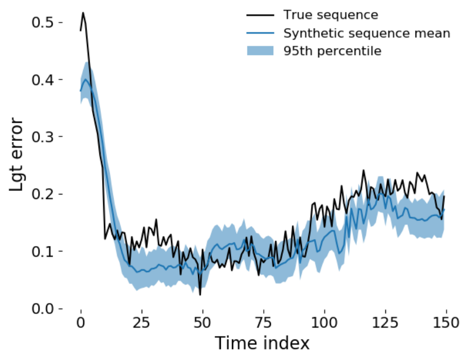
Table 1 shows that the RC-GAN beat the AIO-HMM in terms of RMSE, but that G yielded the best results. However, G performed very poorly in terms of JSd overall, not being able to model the underlying distribution. We conclude that the RC-GAN is the best performing model. By using many non-linear transformations and internal memory, the it is able to learn rich representations and model long-term dependencies better than the other models.



(a)



(b)



(c)

Figure 2. Time series describing the long. position error for three tracked objects from the test set (black) together with the mean (blue) and 95th percentile (filled blue) of the 100 generated time series from the RC-GAN. The vertical axis has been re-scaled due to sensitive data.

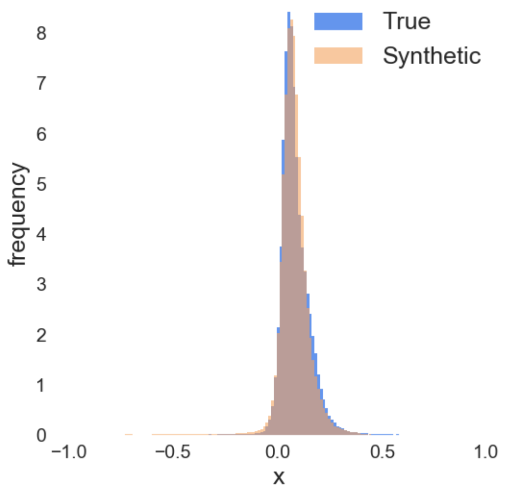


Figure 3. Histogram of all time series from the test set (blue) together with the the generated time series from the RC-GAN (orange) for each test sequence. The vertical axis shows a normalised frequency and the horizontal axis has been re-scaled due to sensitive data.

Table 1. Table showing the results for the AIO-HMM, G and the RC-GAN.

Model	JSd	1^{st} diff JSd	RMSE
AIO-HMM	0.13	0.15	0.67
G (MAE)	0.342	0.550	**0.392**
RC-GAN	**0.082**	**0.110**	0.503

7. Conclusions

In this paper, a Recurrent Conditional Generative Adversarial Network (RC-GAN) has been proposed for modelling real valued time series describing sensor outputs that are used in autonomous driving. The RC-GAN is able to handle time series of arbitrary length and also having the ability to tune the noise levels to the specific data distribution that is wished to be learned. As it is possible to run the model on arbitrary long sequences it is also possible to train the network on data sets containing sequences of different lengths, which we do in this paper. The proposed model is able to in a stable way generate time series with long-term temporal dependencies using internal memory and learns the underlying real data distribution. In particular, we yield better results for generating time series for sensor outputs than those reported in previous work by a great margin.

References

Briët, J. and Harremoës, P. Properties of classical and quantum jensen-shannon divergence. *Physical review A*, 79(5):052311, 2009.

Donahue, C., McAuley, J., and Puckette, M. Synthesizing audio with generative adversarial networks. *CoRR*, abs/1802.04208, 2018. URL `http://arxiv.org/abs/1802.04208`.

Esteban, C., Hyland, S. L., and Rätsch, G. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633*, 2017.

Florbäck, J., Tornberg, L., and Mohammadiha, N. Offline object matching and evaluation process for verification of autonomous driving. In *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, pp. 107–112. IEEE, 2016.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

Kalra, N. and Paddock, S. M. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 94:182–193, 2016.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A. P., Tejani, A., Totz, J., Wang, Z., et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, volume 2, pp. 4, 2017.

Listo Zec, E., Mohammadiha, N., and Schliep, A. Statistical sensor modelling for autonomous driving using autoregressive input-output hmms. In *The 21st IEEE International Conference on Intelligent Transportation Systems*, 2018.

Mogren, O. C-rnn-gan: Continuous recurrent neural networks with adversarial training. *arXiv preprint arXiv:1611.09904*, 2016.

Reed, S. E., Akata, Z., Mohan, S., Tenka, S., Schiele, B., and Lee, H. Learning what and where to draw. In *Advances in Neural Information Processing Systems*, pp. 217–225, 2016.

Theis, L., Oord, A. v. d., and Bethge, M. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

Yu, L., Zhang, W., Wang, J., and Yu, Y. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, pp. 2852–2858, 2017.