Which deep artifical neural network architecture to use for anomaly detection in Mobile Robots kinematic data?

Oliver Rettig[1], Silvan Müller[1], Marcus Strand[1], Darko Katic[2]

[1] Baden-Wuerttemberg Cooperative State University, Department for Computer Science, D-76133 Karlsruhe, Germany,
oliver.rettig@dhbw-karlsruhe.de
[2] ArtiMinds Robotics GmbH, D-76139 Karlsruhe, Germany

Abstract. Small humps on the floor go beyond the detectable scope of laser scanners and are therefore not integrated into SLAM based maps of mobile robots. However, even such small irregularities can have a tremendous effect on the robot's stability and the path quality. As a basis to develop anomaly detection algorithms, kinematics data is collected exemplarily for an overrun of a cable channel and a bulb plate. A recurrent neuronal network (RNN), based on the autoencoder principle, could be trained successfully with this data. The described RNN architecture looks promising to be used for realtime anomaly detection and also to quantify path quality.

Keywords: neural networks, DL4J, anomaly detection, inertial sensor data, mobile robotics, deep learning

1 Introduction

The navigation of mobile robots typically relies on laser scanner data. Small humps on the floor, e.g. cable channels, doorsills, floor unevenness or other environmental anomalies go beyond its detectable scope. Typically only a 2D map of the environment e.g. 10cm over ground can be established. However, even such small irregularities can have a tremendous effect on the robot's stability and the path quality. Induced vibrations can impact cargo or can reduce the storage life of the robot or its mechanical components.

The new idea of our project is to seek to integrate the detection of small anomalies into dynamic adaptation during the execution of a path and into path planning itself. This should be done based on acceleration data, which can be collected simple and inexpensive by inertial sensors.

Commercial mobile platforms like the Mir-100 allow the definition of driving routes by defining manually a few target points in the map. Then, subsequent path planning is done automatically considering several boundary conditions, e.g. distances to walls. Such a map based path planning can be extended by dynamic path planning in order to adjust to temporary changes in the environment [1]. By driving around or stopping in front of unpredicted and potentially dynamic obstacles collisions can be avoided.

J. Beyerer et al. (Eds.), *Machine Learning for Cyber Physical Systems*, Technologien für die intelligente Automation 9, https://doi.org/10.1007/978-3-662-58485-9_7

2 Methodology

In robotics typically high-dimensional sensory data with application specific configurations are in use. To make an anomaly detection component reusable without expensive adaptions from specialists, it is desirable to base on a flexible architecture (one or many input channels) and not to use much domain knowledge about the data. This and the need to work with streaming data to find anomalous subsequences instead only single outliers, quantifiable by a score, exclude many anomaly detection methods available in the literature.

On the other side, artificial neuronal networks in general have been used to solve a large range of problems in the field of robotics processing [2] particularly, deep-learning networks are identifyed as the leading breakthrough technique in the field of mobile robots [3]. They might be used to overcome important challenges in perception and control of mobile robots. For example in [5, 6] a novelty detection in visual data to analyze the robot's environment is described.

In [13] we have shown that a specific deep neural network (DNN) based autoencoder allow for a robust and easily expandable implementation of anomaly detection in kinematic data but which architecture should we use?

There are several approaches. A common way is to train a neuronal network with non anomalous data to be able to predict the next few time frames in the timeseries, based on the current and past values. Then the test data can be compared with the predicted data and the prediction error gives an indication of anomaly [4].

A further class of unsupervised methods combines recurrent neural networks with an encoder/ decoder used as a reconstruction model, where some form of reconstruction error is used, as a score measure of anomaly. The so called autoencoders are trained to reconstruct the normal time-series and it is assumed, that such a model would do badly to reconstruct anomalies, having not seen during training [4].

A newer variant of the autoencoder architecture is the variational autoencoder (VAE) introduced in [7, 8] and amongst others used for anomaly detection [9]. It is based on a reconstruction probability instead a reconstruction error, which should be a more objective anomaly measure. To take into account the temporal structure of timeseries in such an architecture, an additional LSTM [11] layer can be preceded.

3 Concept

The bigger aim of the project behind this paper is to make the usage of mobile robots more robust and flexible by dynamic adaptions to a changing environment. This paper extends the work in [13], which describes in detail the kinematics of the commercially available mobile platform Mir-100 during overrun of a cable channel as a model for an environmental anomaly. Takeoffs are happening particular strong for the rear wheels as a product of the front and the drive wheels already past the cable channel and therefore pulling is more

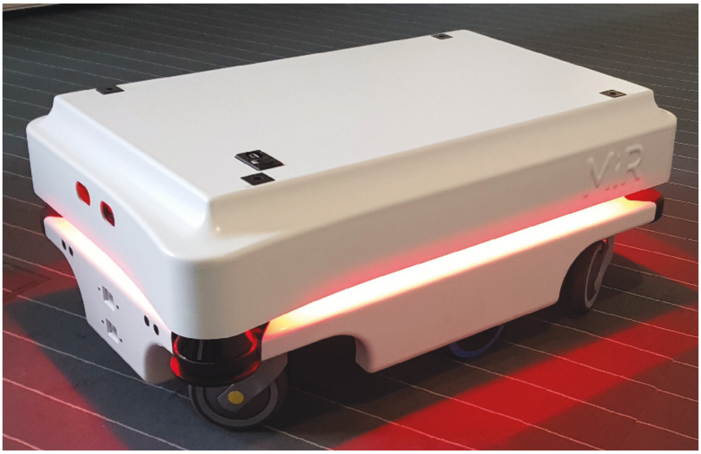

Fig. 1. Commercially available MiR-100 mobile platform.

effectively. To avoid a damage of the platform or its cargo the idea is to detect the overrun of the front wheels as an anomaly in realtime and to slow down the mobile platform before the rear wheels reach the cable channel.

The measurements described in [13] are done with high precision by a marker based optical system to have a "gold standard". This dataset is also used to train the DNNs presented in this paper.

4 Experiments

Two DNNs are implemented based on DL4J, an open sourced, industry-focused, commercially supported distributed deep-learning framework, which supports multiple CPUs and GPUs.

Furthermore architectures based on a convolutional layer to extract features along the time axis and fed them into a recurrent or dense layer are tried.

The first tested architecture consists of a sequence of four network layers, three of type LSTM [12] with 64, 256 and 100 nodes and hyperbolic tangent as activation function, followed by a dense layer with 100 nodes and linear activation. For fitting the weights, mean squared error is chosen as loss function and RMSPROP, which keeps a moving average of the squared gradient for each weight, as optimizer.

The second architecture consists of six network layers. The first of type LSTM [12] with one input node and 100 output nodes, followed by an variational autoencoder (VAE) introduced in [7, 8] and amongst others used for anomaly detection [9]. It has two encoder- and two decoder-layers, 256 nodes each. The end of the sequence builds a dense output layer.

Both DNNs are trained with vertical acceleration data from the reference dataset which was collected in high precision by a marker based optical system during driving a mobile platform Mir-100 (Fig. 1) in a gait- and motion analysis lab. Details of the dataset and its aquisition is described in [13]. Three trials are arbitrary chosen to build a validation set.

The DNNs are trained with the remaining 24 example trials with about 15000 time frames each. Only the sections of the trial without the overruns of the cable channel are included in the training set. Over each trial a time window of width 100 frames is moved step by step and the resulting 100 * trial length sequences are mixed up to build the training sequence. To normalize the data and make it more suitable as input for the DNN the mean is subtracted and a division by the standard deviation is done.

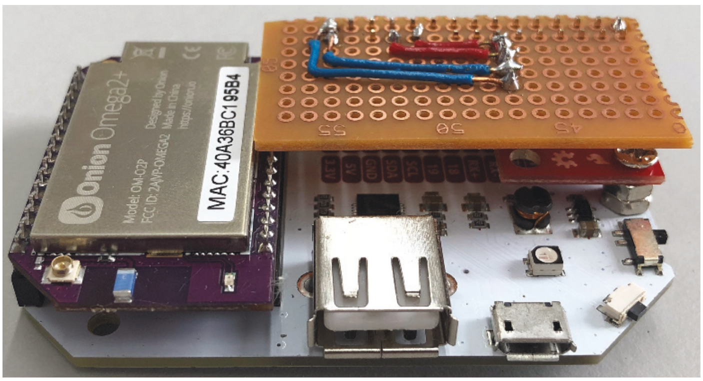

Fig. 2. Inertial measurement unit MPU 9250 + Onion Omega2.

Further three test trials with acceleration data (sampling rate 120Hz) are collected from an inertial measurement unit MPU 9250 (Inven Sense) connected via I2C to a Omega2 module (Onion, Fig. 2) and mounted on the mobile platform. To test the DNNs the data is saved in csv files. In principle the data can be streamed via WiFi to an external laptop, which also collects the position data of the mobile platform via the MiRs REST-API.

Vertical acceleration data is collected for three test trials during driving the robot in a corridor with full speed. A cable channel (Fig. 3) is overrun in the middle of the trial.

5 Results

Training of LSTM based autoencoder and the VAE (4) both converges well with a batch size of 50 and a learning rate of 0.2. Loss function values after training with 1 and after 5 epochs are 4.686 and 1.154 for the LSTM layers based autoencoder and 0.619 and 0.039 for the VAE. The values show no differences between the three test trials (optical marker based measurements) for the shown digits.

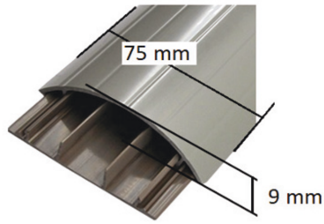

Fig. 3. A cable channel as an anomaly model.

Reconstructed non anomalous data looks very similar in both cases and the overruns of the cable channels are detected clearly as anomaly in all (validation- an inertial sensor based test trials) cases. Fig. 5 shows the difference between original and the predicted/reconstructed data for non anomalous data. The data was normalized to one for the complete trial inclusive anomalous data. That is why the values for non anomalous data in Fig. 5 are so small. Fig. 6 shows a part of the same trial with anomalous data. The three peaks correspond with the overrun of the front-, drive- and rear-wheels. The detections work fine too for inertial sensor based test trials although the DNNs are trained with the marker based optical high precision lab data only.

The approach with a convolutional layer based architecture has no success until now.

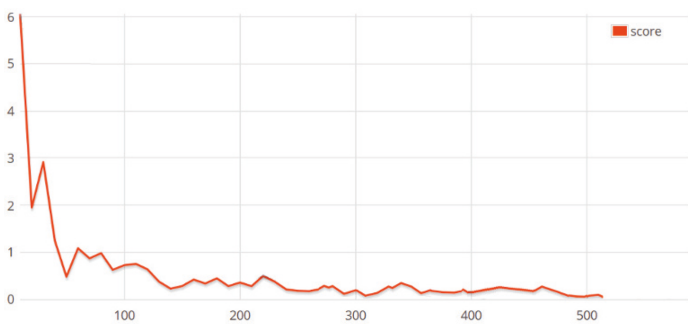

Fig. 4. Score (value of the loss function) over the current minibatch (x-axis), during training of the VAE.

6 Discussion

Anomaly detection works fine for both tested DNN architectures but training of the VAE converges faster and to smaller loss function values which can be an advantage.

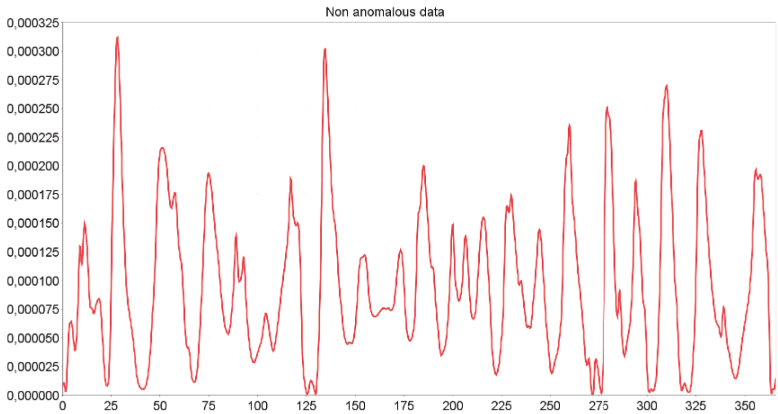

Fig. 5. Normalized anomaly score (predicted minus original acceleration in z-direction) of the VAE based autoencoder; non anomalous data.

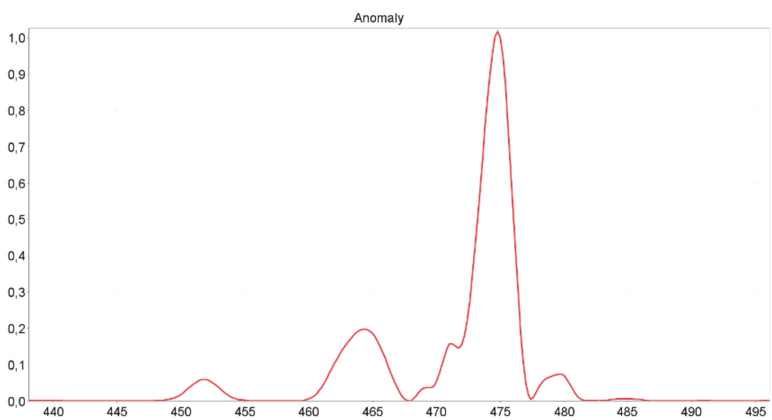

Fig. 6. Normalized anomaly score (predicted minus original acceleration in z-direction) of the VAE based autoencoder; overrun of a cable channel. The three peaks correspond with the overrun of the front-, drive- and rear-wheels. The peak corresponding to the rear-wheels is the biggest one.

These positive results should not hide the fact that a neural net application often needs more care and expenditure in its configuration than an explicit formulated algorithm. Neural nets always come along with the risk to learn hidden but unwanted rules by so called overfitting. In practice you can meet this by a number of arrangements. Carfully choosen architecture details, e.g. for the variational autoencoder used for this project the count of hidden nodes is set higher than the count of input/output nodes. This helps a lot against overfitting. Furthermore you can use so called data augmentation techniques, if the training data set is not divers enough or too small. To be sure that the DNN learns the concrete paths of the training data as normal, we cut the complete movement paths into pieces and create the training set with an random sequence of these pieces.

If the configuration is such sensitive, why to use a neural net et all? The overrun of the cable channel produces a time window with spikes. With a simple threshold spike detector anomaly detection could be achieved with less effort. Furthermore, this could have the additional advantage that the time threshold for spiky data considered as anomalous, can be defined explicitly, so that the concrete mobil platform is meaningful affected. If only 1D acceleration data is available this can be the better approach.

However, if multichannel data is available e.g. from multiple 3d-acceleration and other sensors in combination and if the algorithm should be robust against single sensor dropouts, the DNN approach is more flexible. It is much easier to train a DNN with a different sensor configuration than to adjust thresholds for multiple sensors and to implement a configuration specific logic to make the system robust against dropouts.

The failure of our convolutional layer approach seems to be caused by a too small training data set.

7 Conclusion and Future Work

The DL4J and its VAE implementation has proved in our project as a production ready framework for anomaly detection in mobile platforms acceleration data. This motivates to implement the newer so called variational recurrent autoencoder (VRAE) [10] based on DL4J. The VRAE extends the VAE and takes into account the dynamic temporal behaviour from the scratch.

The next step is to establish a multichannel approach with three or more 3D acceleration sensors and an optimization of the hyper parameters. For this purpose the DL4J provides the the promising so called Arbiter API.

References

1. Meyer J., Filliat D.: Map-based navigation in mobile robots: II A review of map-learning and path-planning strategies. Cognitive Systems Research 4 283-317 (2003)
2. Gamboa, J. C. B.: Deep Learning for Time-Series Analysis. arXiv preprint arXiv:1701.01887. (2017)

3. Tai L., Liu M.: Deep-learning in mobile robotics-from perception to control systems: A survey on why and why not. arXiv:1612.07139. (2016)
4. Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., Shroff, G.: LSTM-based encoder-decoder for multi-sensor anomaly detection. arXiv preprint arXiv:1607.00148. (2016)
5. Neto, H. V., Nehmzow, U.: Real-time automated visual inspection using mobile robots. Journal of Intelligent and Robotic Systems, 49(3), 293-307 (2007)
6. Sofman, B., Neuman, B., Stentz, A., Bagnell, J. A.: Anytime online novelty and change detection for mobile robots. Journal of Field Robotics, 28(4), 589-618 (2011)
7. Kingma, D. P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114. (2013)
8. Rezende, D. J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. arXiv preprint arXiv:1401.4082. (2014)
9. Sölch, M., Bayer, J., Ludersdorfer, M., van der Smagt, P.: Variational inference for on-line anomaly detection in high-dimensional timeseries. arXiv preprint arXiv:1602.07109. (2016)
10. Fabius, O., van Amersfoort, J. R.: Variational recurrent auto-encoders. arXiv preprint arXiv:1412.6581 (2014)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation, 9(8), 1735-1780 (1997)
12. Graves, A.: Supervised sequence labelling with recurrent neural networks. Studies in Computational Intelligence 385 (2012)
13. Rettig, O., Mller, S., Strand, M., Katic, D.: Unsupervised Hump Detection for Mobile Robots Based on Kinematic Measurements and Deep-Learning Based Autoencoder. IAS-15 (http://www.ias-15.org) 2018 (submitted and accepted)