Superseeding Public Transport Timetables With AI

Robert Bendun
robert@bendun.cc
Adam Mickiewicz University
Poznań, Poland

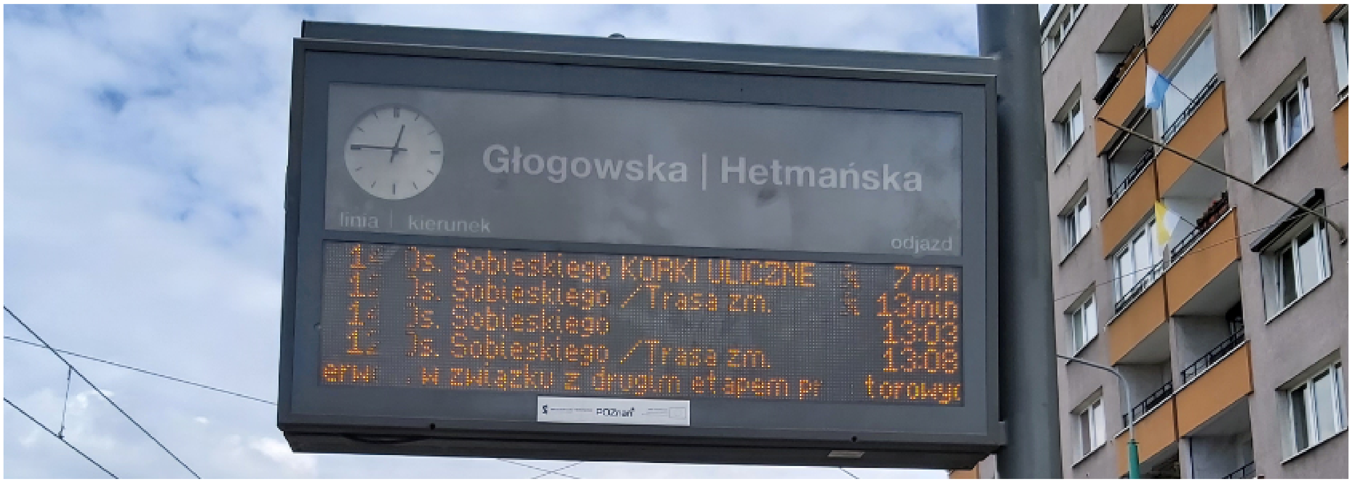

Figure 1. Information table with nearest departures

Abstract

A clear and well-documented LaTeX document is presented as an article formatted for publication by ACM in a conference proceedings or journal publication. Based on the "acmart" document class, this article presents and explains many of the common variations, as well as many of the formatting elements an author may use in the preparation of the documentation of their work.

CCS Concepts: • **Social and professional topics → Automation**.

Keywords: neural networks, public transport

ACM Reference Format:
Robert Bendun. 2023. Superseeding Public Transport Timetables With AI. In *Proceedings of The conference of the ACH Special Interest Group on Harry Quartet Bovik (SIGBOVIK '23).* ACM, New York, NY, USA, 2 pages. https://doi.org/XXXXXXX.XXXXXXX

SIGBOVIK '23, April 1, 2023, Pittsburgh, Pennsylvania

ACM ISBN 978-1-4503-XXXX-X/18/06…$0
https://doi.org/XXXXXXX.XXXXXXX

1 Disclaimer

This article is written for educational purpose. Any and all opinions and information listed in this article should be considered as not representative of me, my university and my employer. Reader discretion is advised.

2 Introduction

Public transportation systems are an essential part of modern city infrastructure, providing reliable and efficient transportation for millions of people every day. However, the reliability of public transport services can often be compromised due to a variety of factors such as traffic congestion, weather conditions, and unexpected events.

Devices showing nearest public transport arrivals on given stop (like one on figure 1), often require Internet access, introducing unnessesary network traffic and increasing city carbon footprint. To combat climate change and reduce noise introduced by network traffic with devices syncing current tram and bus positions, change is required.

We propose to replace network-enabled embeded systems with networkless devices that contain prediction model described by this paper. By reducing complexity of device by reduction of it's capabilities we can reduce production costs and availability of new devices. We can additionally reduce device shown on figure 1 by displaying only the direction for the nearest tram or bus arrival. This allows to reduce problem into multiclass classification.

3 Related work

Most of AI usage inside the public transportation context is concerned with optimization of schedules. Searching for work that uses AI in interaction between passenger and public transport system is rather difficult.

Strongest connection can be found with general AI in public transport articles, especially ones overwieving applications of AI in public transport [4].

4 Method

4.1 Dataset

Dataset from which train and test data were created is publicly accessible public transport schedule information of ZTM Poznań [2]. Due to storage limit, few files from last few months are selected, resulting in 400 MiB initial dataset size. Then data is transformed from CSV format to TSV format, which is more suitable for standard shell text utilities consumption. Utility is written in Go for both performance and ease of use thanks to builtin CSV parser [1].

Next, data is normalized using hand-written tool in C++: data notation is changed from `HH:MM` format to floating point representation when span from `00:00` to `23:59` is mapped to span $[0, 1]$. If row doesn't contain all required information then it's rejected. All columns that are not nessesary are removed. This results in 176 MiB (originally 400 MiB).

Training, validation and test data are extracted from normalized file using scikit-learn [5] function `train_test_split`. All classes in dataset are extracted from normalized dataset using standard POSIX utilities: `cut`, `uniq`, `sort`.

4.2 Model

Model is implemented using Tensorflow [3] framework, both to develop and evaluate. Design of the model is driven by the computational capabilities of Lenovo Thinkpad x270 with i5-7300U processor and 8GB of RAM.

Model is constructed as shown below:

```
from tf.keras import Sequential
from tf.keras.leyers import Input, Dense

model = Sequential([
    Input(shape=(2,)),
    Dense(4*num_classes, activation='relu'),
    Dense(4*num_classes, activation='relu'),
    Dense(4*num_classes, activation='relu'),
    Dense(num_classes, activation='softmax')
])
```

Notable used activation function is softmax, defined as: $\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$ for $i = 1, \ldots, K$ and $z = (z_1, \ldots, z_K)$.

5 Results

Accuracy while training for 2 epochs, epoch size is 1024.

Table 1. Accuracy of trained model

Accuracy
0.20598010947207804
0.20598010947207804
0.18560214941090175
0.19518890350138754
0.19516771079968306

Further evaluation and model training is required.

6 Conclusions

Due to available computational power (or lack there off) any conclusions about the solution are limited. However, with model defined as above we can efficiently compress information - timetables storing required information for a month are 62 MiB and stored model occupies only 19 MiB. Due to greater space-efficiency of AI powered solution, test deployment may be tested in the near future.

References

[1] [n. d.]. Go CSV parser. https://pkg.go.dev/encoding/csv

[2] 2023. ZTM timetables public data. https://www.ztm.poznan.pl/pl/dla-deweloperow/index

[3] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/ Software available from tensorflow.org.

[4] Rusul Abduljabbar, Hussein Dia, Sohani Liyanage, and Saeed Asadi Bagloee. 2019. Applications of Artificial Intelligence in Transport: An Overview. *Sustainability* 11, 1 (2019). https://doi.org/10.3390/su11010189

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009