Machine Learning for Intelligent Mobile User Interfaces using Keras

Huy Viet Le
University of Stuttgart
Stuttgart, Germany
huy.le@vis.uni-stuttgart.de

Sven Mayer
University of Stuttgart
Stuttgart, Germany
sven.mayer@vis.uni-stuttgart.de

Abdallah El Ali
Centrum Wiskunde & Informatica (CWI)
Amsterdam, The Netherlands
abdallah.el.ali@cwi.nl

Niels Henze
University of Regensburg
Regensburg, Germany
niels.henze@ur.de

Copyright is held by the author(s)/owner(s).
Presented at the ACM International Conference on Human-Computer Interaction with Mobile Devices and Services.
MobileHCI'18, September 03-06, 2017, Barcelona, Spain.
© 2018.

Abstract
High-level APIs such as *Keras* facilitate the development of deep learning models through a simple interface and enable users to train neural networks within a few lines of code. Building on top of *TensorFlow*, trained models can be exported and run efficiently on mobile devices. This enables a wide range of opportunities for researchers and developers. In this tutorial, we teach attendees three basic steps to run neural networks on a mobile phone: Firstly, we will teach how to develop neural network architectures and train them with *Keras* based on the *TensorFlow* backend. Secondly, we show the process to run the trained models on a mobile phone. In the final part, we demonstrate how to perform *Human Activity Recognition* using existing mobile device sensor datasets.

Author Keywords
Machine learning; classification; regression; supervised learning; tensorflow; keras; mobile device.

ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous

Introduction
We currently witness the third wave of machine learning. In contrast to the previous waves, current machine learn-

ing research demonstrated impressive performance for a very wide range of tasks. Recent success in machine learning was mainly enabled by combining new training algorithms, new network architectures and moving the training on graphic cards. Combined, these three aspects enable to train models that not only outperform previous approaches but also enable new application areas. Machine learning models are on par with humans or even demonstrate superhuman performance for a number of tasks, including playing Go[1], playing Atari games [12], classifying images[2], or to determine the location where an image was taken [14].

The human-computer interaction (HCI) community used machine learning for a very wide range of use cases. Amongst others, this includes classification of pro-eating disorder [2], authentication [3], touchscreen latency reduction [5, 6, 9], estimating finger orientation [11], enabling palm interaction [7, 8], workout trainer [13], chatbots [15], or using users' emotional facial expressions to enable more natural mobile interaction [4]. Training and using advanced machine learning models recently became much easier due to a variety of libraries, e.g., Torch, Theano, and *TensorFlow*. These libraries are used by researchers from academia and industry but also accessible for practitioners.

With increasing processing power, it became possible to train increasingly complex machine learning models. Luckily processing power is mainly needed during training. A unique feature of *TensorFlow* is the possibility to reduce the size of a trained model and compile it for deployment. In particular, it is possible to run models efficiently on mobile devices. This makes *TensorFlow* especially exciting for mobile HCI researchers as it enables using powerful machine learning models directly on end user devices.

In this tutorial, attendees will learn the basics to develop neural networks and train them using *Keras* based on the *TensorFlow* backend. We will show export and size reduction features of *TensorFlow Mobile* to run trained models on a mobile phone. *TensorFlow* is an open source library for a wide range of machine learning algorithms and includes *Keras* as a simple interface to develop neural networks. *TensorFlow* can move data between CPUs and GPUs to efficiently manipulate them based on CUDA[3] and cuDNN[4].

Covered Topics

We will teach how to train a model using *TensorFlow* version 1.5+ and the integrated Keras API on Python 3.6. We will focus on all necessary steps to use a trained model within an Android application. This tutorial covers neural networks for two purposes, namely classification and regression tasks. The focus lies primarily on supervised learning which enables the models to be used for novel interaction techniques as shown in previous HCI work. For each topic, we provide multiple exercises that attendees will solve in group work and with the support of the instructors.

Intended Audience

Programming knowledge required while machine learning fundamentals can be beneficial. We will use Python in the course to use the *Keras* API. Further, knowledge of state of the art machine learning concepts such as convolutional neural networks is helpful to apply the presented concepts.

[1] David Silver and Demis Hassabis on AlphaGo: Mastering the ancient game of Go with Machine Learning https://research.googleblog.com/2016/01/alphago-mastering-ancient-game-of-go.html

[2] Blog post Andrej Karpathy about what I learned from competing against a ConvNet on ImageNet http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/

[3] CUDA is a parallel computing platform and programming model invented by NVIDIA https://developer.nvidia.com/cuda-zone

[4] CUDA Deep Neural Network library (cuDNN) is a GPU-accelerated library of primitives for deep neural networks. https://developer.nvidia.com/cudnn

Time	Topic
10	Introduction of the agenda, the organizers and the participants
30	Overview of machine learning and recent advances in the field, covering supervised & unsupervised learning, classification & regression, *Keras*, a typical tool chain, and neural networks
50	Hands-on introduction to Jupyter. Participants train a neural network using provided Jupyter notebooks and explore the effect of different hyperparameters using the MNIST data set [10] for handwritten digit recognition.
30	Discussion of performance metrics for classification and regression as well as cost functions
10	Overview of the process to bring *TensorFlow* models to Android devices
30	Participants port their trained MNIST model to Android devices using provided code samples which will result in a handwritten digit keyboard.
30	Overview of body-worn *Human Activity Recognition* research [1], applications for HCI, unique characteristics of time-series modeling, and how to perform supervised learning on time series accelerometer data.
60	Participants train a neural network using provided Jupyter notebooks and experiment with time series accelerometer data using the USC data set [16] for human activity recognition.
30	Participants port their trained *Human Activity Recognition* model to Android devices using provided code samples which will result in a model that recognizes user activities.
20	Wrap-Up of the tutorial, Q&A, and pointers to further directions

Table 1: The structure of the tutorial that mixes discussion of basic concepts and hands-on work. The table shows a rough estimate of the duration in minutes of the different parts based on organizers previous experience.

Materials

Attendees receive accounts to run Python code on our server during the tutorial. We further provide Jupyter notebooks that comprise plug & play examples for a classification and regression task. Participants are provided with the option to download these notebooks (including their changes during the course) on their own machine. These notebooks are designed so that they can be readily modified to run the classification and regression task with the attendee's own data set on their machine for future projects. We further provide instructions, scripts and a demo Android project to run the trained model on an Android device.

Procedure

The tutorial is designed as a course that guides participants through the process of designing the architecture of a neural network, training the model, and deploying it on mobile devices (see Table 1). A full day (five hour) workshop will be sufficient to give participants first hands-on insights into using machine learning for mobile devices.

The course mixes introductions to the individual aspects and hands-on parts that enable participants to explore the presented concepts themselves. Thereby, participants can directly apply the introduced concepts on their own computers. To reduce friction caused by installing the required toolchain on participants' computers, we will prepare Jupyter notebooks for participants to run on our server.

Further topics that would be discussed in a full-day course include, preparing participants' own data sets, advanced network architectures and tools, hands-on exploration of regression, and unsupervised learning.

REFERENCES

1. Bulling, A., Blanke, U., and Schiele, B. A Tutorial on Human Activity Recognition Using Body-worn Inertial Sensors. *ACM Comput. Surv.* (2014). http://dx.doi.org/10.1145/2499621
2. Chancellor, S., Kalantidis, Y., Pater, J. A., De Choudhury, M., and Shamma, D. A. Multimodal Classification of Moderated Online Pro-Eating Disorder Content. In Proc. of CHI '17. http://dx.doi.org/10.1145/3025453.3025985
3. Draffin, B., Zhu, J., and Zhang, J. KeySens: Passive User Authentication through Micro-behavior Modeling of Soft Keyboard Interaction. In Proc. of MobiCASE 2014. http://dx.doi.org/10.1007/978-3-319-05452-0_14
4. El Ali, A., Wallbaum, T., Wasmann, M., Heuten, W., and Boll, S. C. Face2Emoji: Using Facial Emotional Expressions to Filter Emojis. In Proc. of CHI EA '17. http://dx.doi.org/10.1145/3027063.3053086
5. Henze, N., Funk, M., and Shirazi, A. S. Software-reduced Touchscreen Latency. In Proc. of MobileHCI '16. http://dx.doi.org/10.1145/2935334.2935381
6. Henze, N., Le, H. V., Mayer, S., and Schwind, V. Improving Software-Reduced Touchscreen Latency. In Proc. of MobileHCI '17. http://dx.doi.org/10.1145/3098279.3122150
7. Le, H. V., Kosch, T., Bader, P., Mayer, S., and Henze, N. PalmTouch: Using the Palm As an Additional Input Modality on Commodity Smartphones. In Proc. of CHI '18. http://dx.doi.org/10.1145/3173574.3173934
8. Le, H. V., Mayer, S., and Henze, N. InfiniTouch: Finger-Aware Interaction on Fully Touch Sensitive Smartphones. In Proc. of UIST '18. http://dx.doi.org/10.1145/3242587.3242605
9. Le, H. V., Schwind, V., Göttlich, P., and Henze, N. PredicTouch: A System to Reduce Touchscreen Latency Using Neural Networks and Inertial Measurement Units. In Proc. of ISS '17. http://dx.doi.org/10.1145/3132272.3134138
10. LeCun, Y. and Cortes, C. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/
11. Mayer, S., Le, H. V., and Henze, N. Estimating the Finger Orientation on Capacitive Touchscreens Using Convolutional Neural Networks. In Proc. of ISS '17. http://dx.doi.org/10.1145/3132272.3134130
12. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. A. Playing Atari with Deep Reinforcement Learning. *arXiv* (2013). http://arxiv.org/abs/1312.5602
13. Park, J. Classifying Weight Training Workouts with Deep Convolutional Neural Networks: A Precedent Study. In Proc. of MobileHCI '16. http://dx.doi.org/10.1145/2957265.2961861
14. Weyand, T., Kostrikov, I., and Philbin, J. PlaNet - Photo Geolocation with Convolutional Neural Networks. In Proc. of ECCV'16. http://dx.doi.org/10.1007/978-3-319-46484-8_3
15. Xu, A., Liu, Z., Guo, Y., Sinha, V., and Akkiraju, R. A New Chatbot for Customer Service on Social Media. In Proc. of CHI '17. http://dx.doi.org/10.1145/3025453.3025496
16. Zhang, M. and Sawchuk, A. A. USC-HAD: A Daily Activity Dataset for Ubiquitous Activity Recognition Using Wearable Sensors. In Proc. of UbiComp '12. http://dx.doi.org/10.1145/2370216.2370438

Instructors

Huy Viet Le is a Ph.D. student at the University of Stuttgart and is part of the SimTech Cluster of Excellence funded by the German research foundation (DFG) as well as the MWK Baden-Württemberg within the Juniorprofessuren-Programm. He is generally interested in human-computer interaction with a focus on mobile interaction methods based on deep learning. Moreover, Huy is working on modeling touch interaction and hand ergonomics to inform the design of novel interaction methods on mobile devices.

Sven Mayer is a Ph.D. student at the University of Stuttgart. He is part of Cluster of Excellence in Simulation Technology, in brief SimTech, funded by the German research foundation (DFG). He is generally interested in all flavors of HCI. Sven's particular interest is in touch interaction and machine learning. His main research interest is modeling of human behavior patterns for interactive system.

Abdallah El Ali is a tenure-track researcher at Centrum Wiskunde & Informatica (CWI) in Amsterdam, working within the Distributed & Interactive Systems (DIS) group. He is broadly interested in mobile human-computer interaction, with recently a focus on in-the-wild emotion sensing and recognition. Specifically, he combines methods from human-computer interaction and machine learning to make sense of and predict user and audience engagement from wearable (physiological) sensor data.

Niels Henze is professor for Media Informatics at the University of Regensburg. In addition, he heads the group for Socio-Cognitive Systems at the University of Stuttgart. Niels is generally interested in mobile human-computer interaction and he is particularly interested in using large-scale studies to develop models of human behavior that can improve the interaction with mobile devices.