# Machine Learning for Intelligent Mobile User Interfaces using TensorFlow

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

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

**Niels Henze**
University of Stuttgart
Stuttgart, Germany
niels.henze@vis.uni-stuttgart.de


*MobileHCI '17*, September 04-07, 2017, Vienna, Austria
ACM 978-1-4503-5075-4/17/09.
https://doi.org/10.1145/3098279.3119915

## Abstract
One key feature of *TensorFlow* includes the possibility to compile the trained model to run efficiently on mobile phones. This enables a wide range of opportunities for researchers and developers. In this tutorial, we teach attendees two basic steps to run neural networks on a mobile phone: Firstly, we will teach how to develop neural network architectures and train them in *TensorFlow*. Secondly, we show the process to run the trained models on a mobile phone.

## Author Keywords
Machine learning; classification; regression; supervised learning; tensorflow; 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 learning 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 enabled new application areas. Machine learning models are on par with humans or even demonstrate super-human performance for a number of tasks, including playing Go[1], playing Atari games [7], classifying images[2], or to determine the location where an image was taken [9].

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 [1], authentication [2], touchscreen latency reduction [3, 4], life-logging [5], workout trainer [8], and chatbots [10]. Training and using advanced machine learning models recently became much easier due to a variety of open libraries, including Torch, Theano, and TensorFlow. These libraries are not only used and developed by researchers from academia and industry but are 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 to use powerful machine learning models directly on end users devices.

In this tutorial, attendees will learn the basics to develop neural networks and train them using *TensorFlow*. Further, we will show how to port the trained model on to a mobile phone using the model size reduction features of *TensorFlow Mobile*. *TensorFlow* is a graph-based open source library for a wide range of machine learning algorithms. The graph structure enables *TensorFlow* to move the data between CPU's and GPU's to efficiently manipulate them. When using GPUs, *TensorFlow* relies on CUDA[3] and cuDNN[4]. While *TensorFlow* is mostly known for its Neural Network abilities, it is also possible to train other models, such as K-Nearest-Neighbor and Linear Models.

## Covered Topics

We will teach how to train a model using *TensorFlow* version 1.2 or later using Python 3.6. We will further 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. Hence, 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

This tutorial requires knowledge of programming. As *TensorFlow*'s Python API is at present the most complete and easiest to use we will use Python throughout the course. Thus, basic knowledge of Python and the fundamentals in machine learning is beneficial. 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] The NVIDIA 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, TensorFlow, a typical tool chain, and neural networks |
| 30 | 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 [6] for handwritten digit recognition. |
| 10 | 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 |
| 20 | Participants port their trained model to Android devices using provided code samples which will result in a handwritten digit keyboard. |
| 10 | Wrap-Up of the tutorial 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 using state of the art tools, training the model, and deploying it on mobile devices (see Table 1). A two-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[5] for participants and prepare accounts on our own server.

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

[5] http://jupyter.org/

## REFERENCES


1. Stevie Chancellor, Yannis Kalantidis, Jessica A Pater, Munmun De Choudhury, and David A Shamma. 2017. Multimodal Classification of Moderated Online Pro-Eating Disorder Content. In *Proceedings of the 35rd Annual ACM Conference on Human Factors in Computing Systems (CHI '17)*.

2. Benjamin Draffin, Jiang Zhu, and Joy Zhang. 2014. KeySens: Passive User Authentication through Micro-behavior Modeling of Soft Keyboard Interaction. In *5th International Conference on Mobile Computing, Applications, and Services (MobiCASE 2014)*, Gérard Memmi and Ulf Blanke (Eds.). Springer International Publishing, Cham, 184–201. DOI: http://dx.doi.org/10.1007/978-3-319-05452-0_14

3. Niels Henze, Markus Funk, and Alireza Sahami Shirazi. 2016. Software-reduced Touchscreen Latency. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '16)*. ACM, New York, NY, USA, 434–441. DOI: http://dx.doi.org/10.1145/2935334.2935381

4. Niels Henze, Huy Viet Le, Sven Mayer, and Valentin Schwind. 2017. Improving Software-Reduced Touchscreen Latency. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '17)*. ACM, New York, NY, USA.

5. Mohammed Korayem, Robert Templeman, Dennis Chen, David Crandall, and Apu Kapadia. 2016. Enhancing Lifelogging Privacy by Detecting Screens. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 4309–4314. DOI: http://dx.doi.org/10.1145/2858036.2858417

6. Yann LeCun and Corinna Cortes. 2010. MNIST handwritten digit database. (2010). http://yann.lecun.com/exdb/mnist/

7. Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. *arXiv preprint arXiv:1312.5602* (2013). http://arxiv.org/abs/1312.5602

8. Jaehyun Park. 2016. Classifying Weight Training Workouts with Deep Convolutional Neural Networks: A Precedent Study. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '16)*. ACM, New York, NY, USA, 854–858. DOI:http://dx.doi.org/10.1145/2957265.2961861

9. Tobias Weyand, Ilya Kostrikov, and James Philbin. 2016. PlaNet - Photo Geolocation with Convolutional Neural Networks. In *Computer Vision – ECCV 2016: 14th European Conference (ECCV'16)*. Springer International Publishing, Cham, 37–55. DOI: http://dx.doi.org/10.1007/978-3-319-46484-8_3

10. Anbang Xu, Zhe Liu, Yufan Guo, Vibha Sinha, and Rama Akkiraju. 2017. A New Chatbot for Customer Service on Social Media. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3506–3510. DOI: http://dx.doi.org/10.1145/3025453.3025496

## Instructors

**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.

**Huy Viet Le** is a Ph.D. student at the University of Stuttgart, funded by the MWK Baden-Württemberg within the Juniorprofessuren-Programm and the German research foundation (DFG) within the SimTech Cluster of Excellence. His research focuses on modeling touch interaction on mobile devices. Furthermore, he is interested in extending the touch input vocabulary on mobile devices using machine learning and external sensors.

**Niels Henze** is an assistant professor at the University of Stuttgart. He heads the group for socio-cognitive systems within the Institute for Visualization and Interactive Systems as well as the SimTech Cluster of Excellence in Simulation Technology. 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.