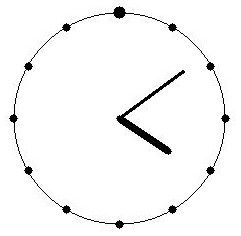
CAS Applied Data Science Modul 5, Peer Consulting

## Marco Bassi Juni 2020

# Projektinhalt

Der Inhalt des Peer Consultings ist das Projekt «Uhren Lesen» von Gerhard Binder; siehe <https://github.com/gerbin56/uhr.git>. G.B. verwendet 3000 Bilder (s/w) des Ziffernblatts einer einfachen Analoguhr mit verschiedenen Zeigerständen, um seinem Neuronalen Netzwerk das Uhrenlesen beizubringen. Bsp:

Die Bilder variieren nach Grösse des Ziffernblatts, und nach Position auf dem Bild.

Die Labels zu den Bildern enthalten pro Zeile Stunde und Minute, die das entsprechende Uhrbild anzeigt. Im obigen Beispiel «4,9». Insgesamt finden sich bei den Labels 713 verschiedene Uhrzeiten. Eine Analoguhr kann – bei einer Auflösung von 1 Minute – 720 verschiedene Zeiten anzeigen.

# Das Neuronale Netzwerk

Das Neuronale Netzwerk (NN) ist mit dem Tensorflow Keras-API implementiert. Es umfasst 3 Ebenen mit je einem Convolutional Layer, einem Maxpooling Layer und einem Batch-Normalisierungslayer. Der erste Convolutional Layer hat 50 Filter, der zweite 100 und der dritte 150 Layer.

Nach diesen drei Schichten folgt nochmals ein Convolutional Layer mit 200 Schichten, danach ein Dropout-Layer (Rate 0.4) und ein Flatten-Layer. Die Activation bei allen vier Convolutional Layern ist ReLu.

Der Output für Stunde und Minute wird getrennt berechnet. Für die Stunde gibt es 2 Fully Connected Layer mit Activation ReLu, und ein Fully Connected Layer mit Activation Softmax, Output-Dimension 12. Und für die Minute hat es analoge Layer, wobei die letzte keine Activation und Output-Dimension 1 hat.

# Was mit gut gefällt

## Wahl der Methode und des NN-Modells

Die Wahl eines NN für die Lösung der Aufgabe ist naheliegend. Der Aufbau des NN hält sich weitgehend an die «Best Practices» gemäss Keras-Dokumentation für die Bildklassifikation. Für jemanden, der mit dem API nicht vertraut ist, ist das sehr hilfreich für das Verständnis.

Die Implementierung des NN ist logisch nachvollziehbar, gut strukturiert und zweckmässig. Auch beim Setzen der Hidden Parameters orientiert sich das NN an den zur Verfügung stehenden Beispielen.

## Manuelle Datenaufbereitung

Das Keras-API bietet Funktionen zum Einlesen, Aufbereiten der Daten und der Trainings- und Test-Datensets. Der Nachteil dieser Funktionen ist, dass sie weitgehend «Black Boxes» sind, d.h. wenn man wissen will, wie diese Funktionen zu ihrem Ergebnis kommen, muss man sich in die Dokumentation einlesen und auch selbst experimentieren.

G.B. implementiert die Datenaufbereitung und die Erstellung der Datensets selbst. Das macht die Ergebnisse direkt nachvollziehbar und überprüfbar.

## Implementierung im Jupyter Notebook

Die Implementierung im Jupyter Notebook ist gut strukturiert und gut nachvollziehbar. Die Inlinedokumentation ist ausreichend und sehr hilfreich für das Verständnis. Dasselbe gilt für die Ausgabe von Zwischenergebnissen, Datenstrukturen und Plots.

# Verbesserungmöglichkeiten und Probleme

## Readme

Im