HOCHSCHULE
RAVENSBURG-WEINGARTEN
UNIVERSITY
OF APPLIED SCIENCES

Applied Deep Learning

Ein Convolutional Neural Network für Spracherkennung
modellieren, trainieren und testen

Melih Öztürk
Matrikelnummer 31447
20. Januar 2020

Gutachter: Mark Schutera

Inhaltsverzeichnis

1 Einleitung **3**
1.1 Motivation 3

2 Datasets and Network **3**
2.1 Informationsgewinnung 3
2.2 Audiodaten Vorverarbeitung 4
2.3 Network 6

3 Fazit **8**

4 Quellen **9**

1 Einleitung

Diese Projektarbeit beschäftigt sich mit der Modellierung eines Convolutional Neural Networks (kurz CNN), welches auf Audiodateien trainiert werden soll, um bestimmte Personen wiederzuerkennen.

1.1 Motivation

Spracherkennung ist weit verbreitet und Methoden wie Mel Frequency Cepstral Coefficients werden oft mit diesem Thema zusammengebracht. Fragen wie “Wie gut lässt sich für Speaker Recognition mit Mel Frequency Cepstral Coefficients in Kombination mit MFCC umsetzen?” sollen mit dieser Projektarbeit geklärt werden.

2 Datasets and Network

Für das Projekt wurden zweierlei Datasets verwendet. Beim ersten wurden aus (Zohar, J., 2016) insgesamt 2000 Audiodateien entnommen. In diesen sind von 4 Personen jeweils 50 Sprachaufnahmen von den Zahlen zwischen 0 und 9. Für diesen Dataset wurde nicht allzuviel Vorverarbeitung der Dateien nötig, welches beim zweiten Dataset nicht der Fall ist.

Beim zweiten Dataset wurde eine Methode implementiert, welche die Informationsgewinnung aus URL Links erlaubt. Bei dieser Methode wurde vorerst nur das herunterladen und extrahieren von Audiodaten aus YouTube Videos implementiert. Hierbei wurden für das Projekt Reden oder Interviews von Politiker gesammelt, um das CNN-Modell zu trainieren.

2.1 Informationsgewinnung

url	strategy	start	end	label
https://www.youtube.com/watch?v=S3C5H-2SqYU	youtube	00:00:15.00	00:00:40.00	angelamerkel
https://www.youtube.com/watch?v=u0KAGFJ76QM	youtube	00:00:01.00	00:00:30.00	aliceweidel
https://www.youtube.com/watch?v=ZtLEcdcd58U	youtube	00:01:10.00	00:01:33.00	karambadiaby
https://www.youtube.com/watch?v=jte-Ch6woAs	youtube	00:00:10.00	00:00:40.00	habeckrobert

Abbildung 1: Beispiel *.csv Datei mit den Spalten “url, strategy, start, end und label”.

Für die Informationsgewinnung wurde eine *.csv Datei definiert, die als Spalten “url, strategy, start, end, label” enthält. Die “url” Spalte enthält wie der Name bereits erwähnt die jeweilige URL einer bestimmten Informationsquelle, welche in diesem Fall nur YouTube Videos sind.

Die Spalte “strategy” enthält als textuelle Darstellung die jeweilige Strategie, welche später im Programmcode implementiert ist, um aus der jeweiligen URL die Informationen zu gewinnen bzw. zu herunterladen und zu extrahieren. In diesem Falls wären alle Reihen mit der Strategie “youtube” befüllt. Die Spalten “start” und “end” bestimmen jeweils wie die gewonnen Information getrimmt bzw. abgeschnitten wird. Hierbei wird das Format HH:MM:SS.MSMS, worin HH für Stunden, MM für Minuten, SS für Sekunden und MS für Millisekunden steht. Die Spalte “label” bestimmt die jeweilige Klassifizierung der Information.

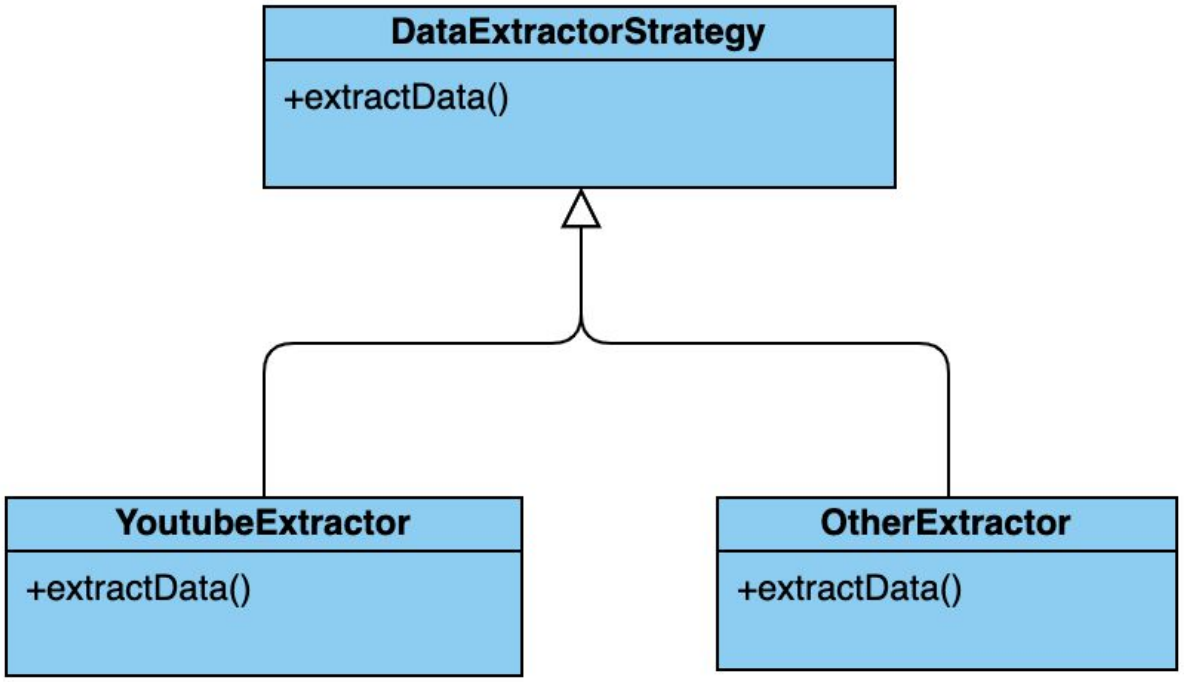


Abbildung 2: DataExtractorStrategy Klassendiagramm.

Im Projektverzeichnis unter “/politicians_sound_project/information_collection.ipynb” ist die Informationsgewinnung implementiert. Dieses enthält die Klasse DataExtractor, welches als einziges als Parameter den Pfad zur *.csv Datei benötigt. Diese Klasse enthält Funktionen zum “Lesen” und “Extrahieren” der Daten aus den Quellen, welche in der *.csv Datei definiert sind. Für das Extrahieren wird je nach ausgewählter Strategie eine bestimmte konkrete Klasse von der Klasse DataExtractorStrategy instanziiert. Hier wurde nur eine Klasse YoutubeExtractor implementiert, welches für das Herunterladen und Ausschneiden von Audiodateien aus YouTube Videos zuständig ist. Diese Klasse wird jedesmal verwendet sobald als “strategy” der Wert “youtube” eingetragen wird.

2.2 Audiodaten Vorverarbeitung

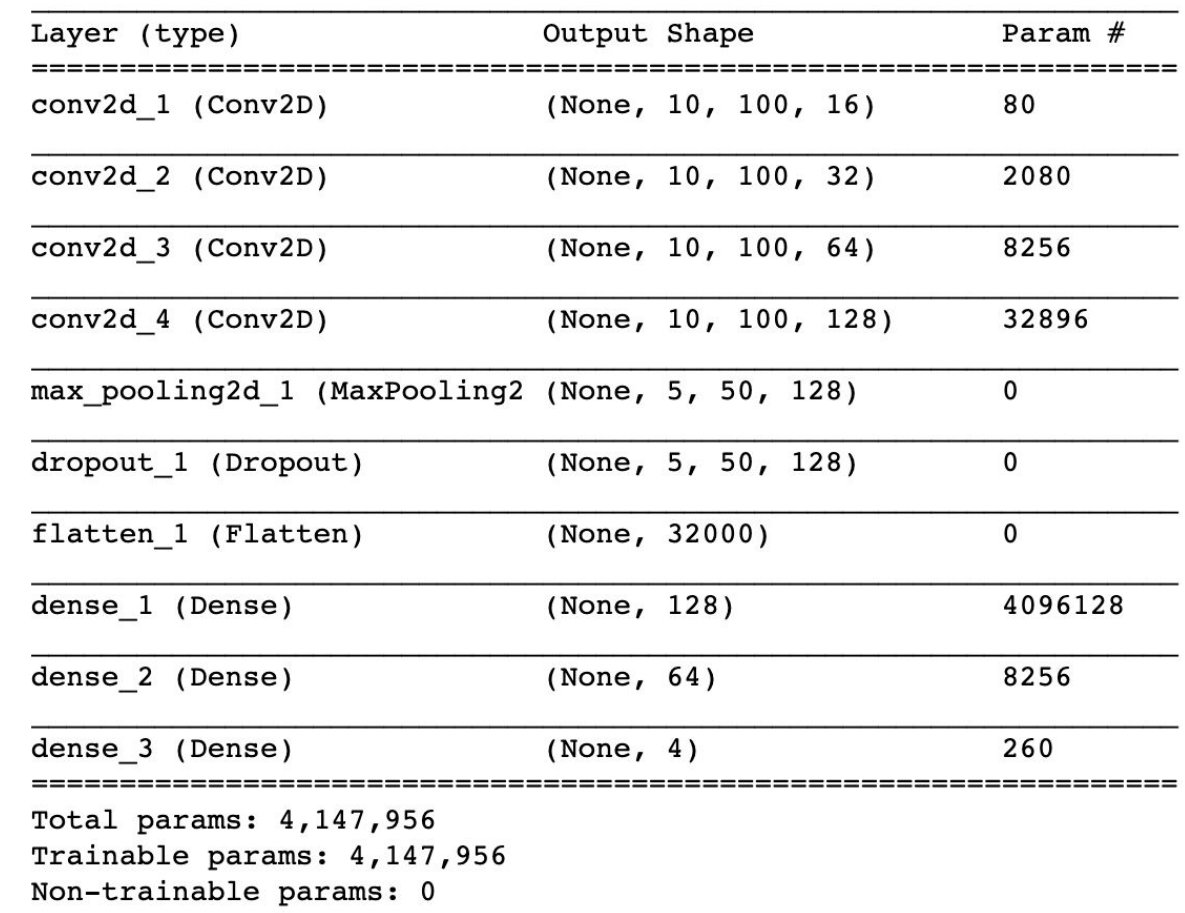
Um das Modell zu trainieren und zu testen müssen die Audiodaten in einer bestimmten Datenstruktur abgelegt werden damit das Convolutional Neural Network damit überhaupt arbeiten kann. Hierfür werden die gespeicherten Audiodateien mit Hilfe von Funktionalitäten aus dem Paket librosa (librosa 0.7.2 documentation) verwendet. Dabei werden die

Audiodateien in Mel Frequency Cepstral Coefficients (kurz MFCC) Daten umgewandelt. MFCC ist eine kompakter Darstellung des Frequenzspektrums (Wikipedia., 2005 May 12).

Abbildung 3: Visualisierung der MFCC Daten

Die Abbildung 3 zeigt die visuelle Darstellung der gewonnenen MFCC Werte aus den Audiodateien. Diese Werte werden verwendet um Test und Trainingsdaten zu generieren. Später zur Klassifikation von neuen Daten wird die bestimmte Audiodatei ausgewählt, welche in MFCC umgewandelt wird und von dieser Audiodatei werden einzelne Stücke als Input in das Modell gegeben.

2.3 Network



Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 10, 100, 16)	80
conv2d_2 (Conv2D)	(None, 10, 100, 32)	2080
conv2d_3 (Conv2D)	(None, 10, 100, 64)	8256
conv2d_4 (Conv2D)	(None, 10, 100, 128)	32896
max_pooling2d_1 (MaxPooling2	(None, 5, 50, 128)	0
dropout_1 (Dropout)	(None, 5, 50, 128)	0
flatten_1 (Flatten)	(None, 32000)	0
dense_1 (Dense)	(None, 128)	4096128
dense_2 (Dense)	(None, 64)	8256
dense_3 (Dense)	(None, 4)	260

Total params: 4,147,956
Trainable params: 4,147,956
Non-trainable params: 0

Abbildung 4: CNN

Die Abbildung 4 zeigt das CNN mit welchem das Dataset mit den gesprochenen Zahlen aus (Zohar J., 2016) trainiert und getestet wurde. Die Eingabeform (10, 100, 16) setzt sich wie folgt zusammen:

1. 10 Feature Werte, welche aus der Extrahierung von MFCC mit Hilfe von librosa Paket gewonnen wird.
2. 100 Spalten aus dem gewonnenen MFCC Wert.
3. 16 als Anfangswert der Neuronenanzahl

Zu Beginn wird die Neuronenanzahl mit einem 2 Dimensional Convolutional Layer zunächst immer verdoppelt bis 128 erreicht wird. Hier wird einmal Max Pooling angewendet, welches ein Pool Size von (2, 2) besitzt. Das heisst für die Heruntertaktung werden jeweils in dem Bereich 2x2 immer die Maximalen Werte übernommen und der Rest verworfen. Nach diesem Layer wird ein Dropout mit dem Wert 0.5 hinzugefügt und ein Flatten Layer, welches sich aus 5*50*128 zusammensetzt. Danach folgen Dense Layer worin die Neuronenanzahl immer halbiert werden. Der letzte Layer ist ein Dense Layer mit der Anzahl der Klassen und der Aktivierungsfunktion Softmax. Softmax wird verwendet, da wie hier eine Multiklassen-Klassifikation haben.

Da das Modell für ein Input nur ein Label vorhersagen soll ist für die Lossfunktion “Categorical Crossentropy” gut geeignet. Für den Optimizer wird “adam” verwendet, welches eine Erweiterung des Stochastik Gradient Descent ist.

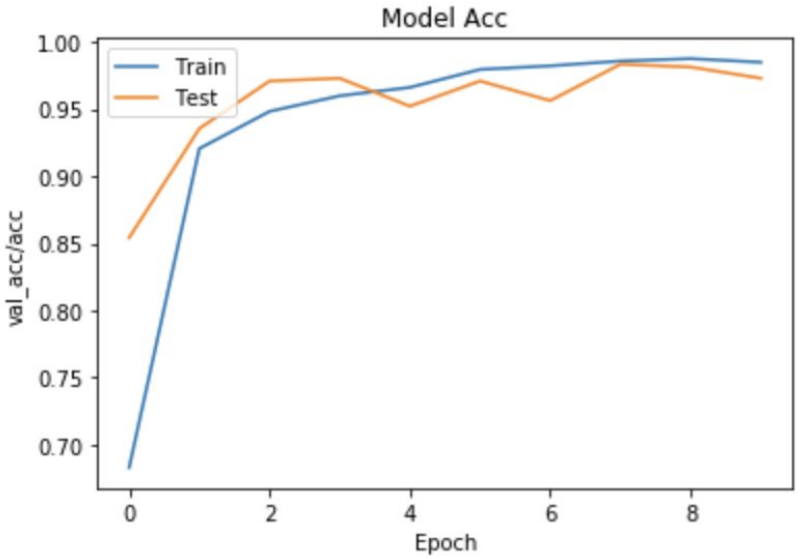


Abbildung 5: Model Accuracy und Model Validation Accuracy

In Abbildung 5 ist die Präzision des Models auf Trainings- und Testdaten zu sehen. In Blau gut zu erkennen ist, wie sich die Genauigkeit der Trainingsdaten gegen 100% nähren aber ab dem 8. Durchlauf etwas wieder entfernen. Die Orangene Linie zeigt die Präzision der Testdaten schwankend. Gegen Ende zwar mit hoher Genauigkeit aber dennoch mit leichtem sinken.

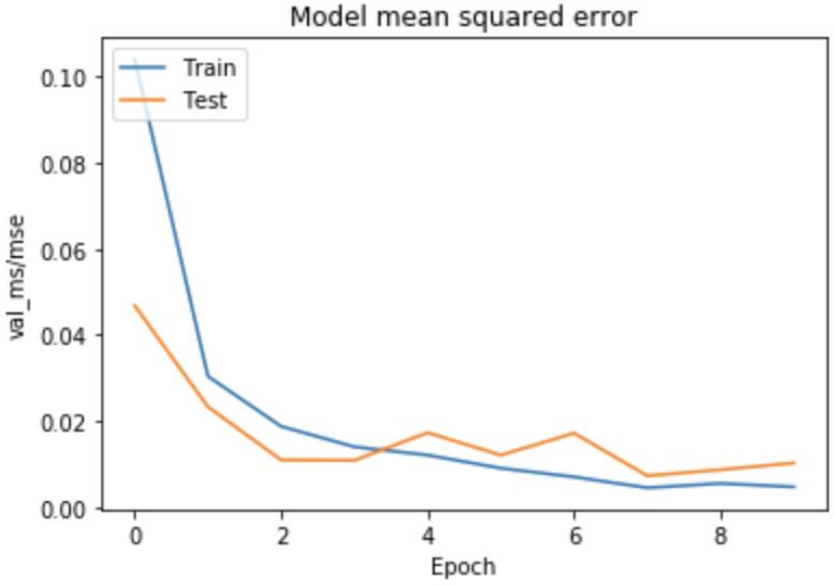


Abbildung 6: Model Mean Squared Error und Model Validation Mean Squared Error

In Abbildung 6 ist der Mean Squared Error (MSE) des Models auf Trainings- und Testdaten zu sehen. In Blau ist zu erkennen, wie sich der Wert der Trainingsdaten gegen 0 nähren. Die Orangene Linie zeigt die Werte der Testdaten schwankend. Gegen Ende zwar mit niedrigem MSE Wert, aber ab dem 5. Durchlauf steigert sich der Wert mit einer ganz leichten konstante.

3 Fazit

Für das Vorverarbeiten von Audio ist ein sehr guter Ansatz Audio Augmentation anzuwenden, welches in (Ma E., 2019 June 10) sehr gut beschrieben ist. Hier wird gezeigt, wie noise (deutsch Lärm) zu Audiodateien hinzugefügt werden kann. Manchmal ist es aber auch nötig, wie im Laufe dieses Projektes aufgefallen ist, dass oftmals noise entfernt werden muss, damit das Modell besser lernen kann. Zusätzlich könnte Verschiebung oder Verzerrung von Audiodateien helfen, dass das Modell besser auf die Daten verallgemeinert lernt. Durch das Projekt wurde ersichtlich das MFCC in Kombination mit CNN ziemlich gut für Speaker Recognition geeignet ist.

4 Quellen

Zohar, J. (2016). Jakobovski/free-spoken-digit-dataset. Retrieved January 20, 2020, from https://github.com/Jakobovski/free-spoken-digit-dataset

librosa 0.7.2 documentation. (n.d.). Retrieved January 20, 2020, from https://librosa.github.io/librosa/generated/librosa.feature.mfcc.html

Wikipedia. (2005, May 12). Mel Frequency Cepstral Coefficients. Retrieved January 20, 2020, from https://de.wikipedia.org/wiki/Mel_Frequency_Cepstral_Coefficients

Ma, E. (2019, June 10). Data Augmentation for Audio. Retrieved January 20, 2020, from https://medium.com/@makcedward/data-augmentation-for-audio-76912b01fdf6