Seminar on the Physics
of Deep Learning

GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Using Convolutional Neural Networks for Speech Recognition

Authors: Jan Cammann, Jorge de Heuvel, Marvin Klingner, David Schlegel
(Dated: August 14, 2017)

In the last decade, Spoken Language Recognition (SLR) has gained interest along with the development of general speech recognition. The classification of languages plays an important role in interpreting and in increasing the performance of speech recognition networks, currently finding more and more implementation in daily-life applications such as smartphones and computers.

In this work, we present a deep learning approach to classify multiple spoken languages with different deep learning network architectures based on convolutional networks. For training and testing our networks, we use the data given by two Topcoder challanges and by VoxForge. For the convolutional neural network architecture, additional preprocessing is required to obtain spectrograms as an input for the network. The aim of this work is to get acquainted with deep learning networks and to analyze the convergence of the network training depending on the network structure. Additionally, we compare our obtained results from testing with language families as a verification by calculating the confusion and correlation matrix from the predicted classes. The main result of our work is that we were dealing with a low diversity in speakers which caused the networks to adapt to the speakers rather than to the language itself. In conclusion one would need to optimize the used datasets with respect to diversity of recordings and speakers.

INTRODUCTION

Spoken language classification is a key in automated language processing. For automated systems in which the user provides vocal input for further processing tasks, it is in most cases crucial to identify the spoken language first. Since spoken languages are complex structures, that depend both on the speaker and on the specific characteristics of the respective language, classical algorithms were rather unsuccessful in classifying spoken languages [1]. This necessitates a different approach for which deep learning networks seem to be a promising and state-of-the-art tool for classifying spoken languages. Additionally, a language classification provides important information for a subsequent language recognition network, which is usually fine-tuned to a specific language.

To highlight the current developement, Google recently released an automated written language recognition for automated text-filling used in mobile devices [2]. This task can also be extended to speech processing devices which currently mostly support language recognition for a specific preset language. Thus, spoken language recognition is an important task in providing a dynamic interface between humans and machines.

In this article, we present an approach to classify languages from a set of a) 5 languages and b) 176 languages, provided by two distinct Topcoder challanges [3, 4] and VoxForge [5] with deep convolutional neural networks. For the implementation, we use Keras with Tensorflow as backend. First, the preprocessing and augmentation of the datasets are explained which is of crucial importance for a high validation accuracy. We present the network architectures that are used to perform the language classification, followed by the obtained results for training and validation accuracy. Furthermore, we compare the similarity of the classified languages for both the Topcoder dataset as well as the VoxForge dataset by analyzing the cross-correlation matrix and the confusion matrix of the prediction vectors to correlate our findings with language families from etymological and historical linguistics.

PREPROCESSING AND AUGMENTATION OF THE DATASETS

With respect to the training of the network, the first task is to collect data from different languages which are of the same length and audio quality. For the networks used here, we use datasets from the two Topcoder challenges and from VoxForge. The datasets contain data in mp3-format (Topcoder) and wav-format (VoxForge) with the small dataset consisting of five

languages with approximately 100 files each and the large consisting of 176 languages with approximately 250 files each, all having the same length of 10 s. The VoxForge data has variable length so that one must choose an audio length for the network. For the final network we chose a length of 4 s.

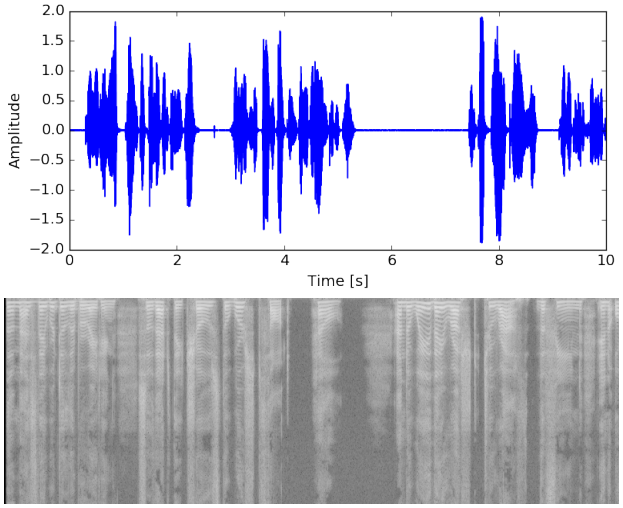

FIG. 1. *Typical shape of the data in the wav-format (top) and the typical shape of the resulting spectrograms saved as a png-image (bottom).*

With this data format containing the amplitude of the audio at different times, we divide the data into slices of 20 ms. Depending on the length of the audio files one can freely choose the length of the image. For the Topcoder Challenges the pictures represent 9 s and for the VoxForge datasets the pictures represent 4 s or 8 s of audio. The offset from which the interval starts can be freely chosen. On the slices of 20 ms, we then apply a Short-Time-Fast-Fourier transform (STFT). This technique has the advantage that the borders of the Fourier-transformed interval get smoothed and converge to zero so that the effects of the cut-off edges have less impact on the resulting spectrum. The resulting data structure is a 2D-array with one axis representing the time and the other axis representing the frequencies. The resulting array is saved in the png-image format. An example spectrogram can be seen in the lower panel of Fig. 1.
As we do not have a large amount of data for each language in our datasets and to prevent the network from learning irrelevant features of the spectrograms, we use two techniques to generate 20 spectrograms from each mp3-file: For each spectrogram we randomly chose an time offset and a coefficient α to transform the frequencies with the following formula:

$$G(f) = \begin{cases} \alpha f & 0 \leq f \leq f_0 \\ \frac{f_{\max} - \alpha f_0}{f_{\max} - f_0}(f - f_0) + \alpha f_0 & f_0 \leq f \leq f_{\max} \end{cases}$$

Here we chose the parameters as $f_0 = 0.9 f_{\max}$ and $\alpha \in [0.9, 1.1]$ randomly. This method is called frequency warping and depending on α stretches or compresses all frequencies for $f < f_0$ and vice versa for all frequencies $f > f_0$. The exact documentation from this method can be found in [6] and [7]. The preprocessing then results in 20 images for every audio file.
The preprocessed data is now split up into a training set and a validation set with a commonly chosen ratio of 80% training data to 20% validation data. All 20 spectrograms from one mp3-file are either in the training or the validation set to prevent the network just learning to recognize differently augmented spectrograms of the same audio file.

STRUCTURE OF THE NETWORKS

The created spectrograms can be viewed as an image of the spoken language. Therefore the classifying task can be treated as an image classification problem. Predominantly in the literature, image classification tasks are coped with using convolutional neural networks, thus we chose a similar approach on the basis of a convolutional neural network architecture for our classification problem.
The exact structure of the model is motivated by [8] who already tried to classify the language sets we used and had a training accuracy of $\approx 97\%$ classifying the large dataset. After some trials, we settled on the following structure: We start with the input layer taking a spectrogram as input and stacked several groups of layer on top, which all consisted of a convolutional layer, a max-pooling layer, and batch normalization afterwards to avoid overfitting [9]. For the small network, we used three and for the large network five of these layers. Because of large success in other networks, a rectified linear unit (ReLu) is used as activation function. On top of the convolutional layers, we apply a fully connected layer (ReLu-Activation) with a varying constant dropout between 30% and 50%, again to avoid overfitting. The final layer is a logic layer with as many units as number of classes with softmax activation function. The structures are sketched in Fig. 2.

For the training of the network, we utilize the categorical crossentropy loss function and a stochastic gradient descent method with 0.003 learning rate and momentum of 0.9 to find the minimum in configuration space. These parameters turned out to be the most successful for convergence of the loss function and the validation score.
The programs were first implemented using the python library Tensorflow and then extended to Keras with Tensorflow backend. The advantages in Keras are the convenient and simple setup of prototype network architectures and the ability to perform continued training and get meta information about the training and validation

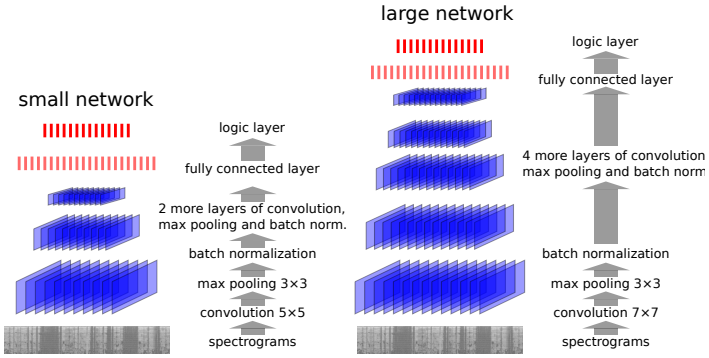

FIG. 2. *Schematic representation of the overall shapes of the networks used for language classification on the small dataset with 5 languages (left) and on the large dataset with 176 languages (right).*

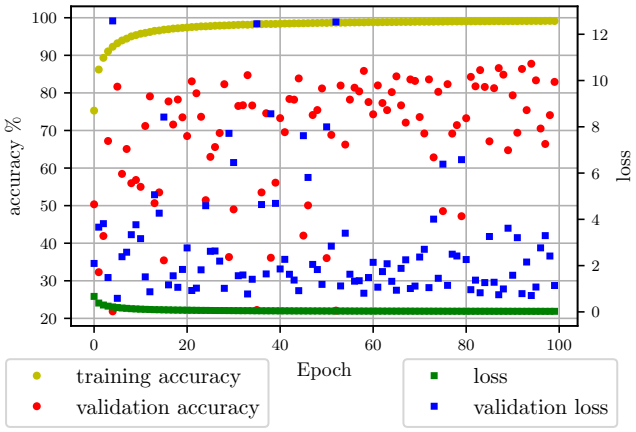

FIG. 3. *Training history for network trained on 4s audio snippets from the VoxForge dataset. Training was done using the Adam optimizer. Since accuracy an loss is only recorded at the end of an epoch and epoch length was here chosen to be very long, the initial convergence is not captured in this plot. We do only see the fluctuating validation accuracy and loss for nearly fully converged weights.*

processes. Additionally, the usage of GPUs is supported, yielding an enormous speed up in comparison to CPUs.

RESULTS

We observe the accuracy of the network being highly dependent on the used dataset. For the Topcoder dataset containing 5 languages, we achieved a validation accuracy of $> 99\%$. For the dataset containing 176 languages, a validation accuracy of $> 95\%$ was obtained. Using the more diverse dataset from `voxforge.org`, we arrived at a maximal validation accuracy of $> 88\%$ by increasing the dataset stepwise. Here we tested different lengths (4 s and 8 s) for the audio files but we could not detect any impact of this feature on our results. As the data files on Voxforge are mostly short audio recordings we chose a length of 4 s for our final network.
The training and validation loss and accuracy for the network trained on `voxforge.org` dataset is shown in Fig. 3. Aso can be seen the training loss and accuracy both converge as expected, with the training accuracy reaching nearly 100%. This already is a sign for overfitting which is a common problem for all of our networks and also in general for convolutional neural networks. The validation loss and accuracy have far more swaying values. The maximum values are at about 88% validation accuracy. The weights of this final network were saved and used for the derivation of the following results.

The networks classifying 5 languages were tested with self recorded audio with a variety of approaches. First, a correctly classified audio file from the validation dataset was chosen and re-recorded by ourself as speakers. We found, that the network misclassified the re-recorded speech in our own voice. Also, freely spoken phrases (in German) were misclassified. To check, in how far our recording equipment influenced the result of our tests, the original file was played back and recorded in front of the microphone to induce artificial noise. Despite of the poorer quality, the network still classified correctly. We investigated the sensitivity of the network to pitch- and speed-altered copies of a correctly classified audio file using audio filters on the original file, with the result of still classifying correctly. Also, this testifies the augmentation algorithm in terms of pitch-altering, since it does not change the predicted language.
It seems, that the network strongly adapts to speakers and their specific phrases of the training dataset. Therefore we suggest the assumption, that the convolutional neural network adapts to specific patterns in the spectrogram, which a certain speaker causes with his voice, independent of what he or she is saying. Furthermore, in splitting the dataset into a validation and training set, spectrograms with the same speaker are contained in both of the datasets, yielding a rather ostensibly high accuracy.

To counteract this phenomenon, a more diverse dataset using audio files from VoxForge was constructed. This dataset despite being larger also features a more diverse collection of speakers using different recording equipment. Since most files provided at VoxForge are fairly short, mostly containing only one sentence, we reduced the audio length to 4s. The original data from VoxForge is also recorded with a lower sample rate (16kHz) in contrast to the data from the Topcoder challenges (44kHz). This leaves us with spectrograms capturing a maximum frequency of 8kHz. This does not affect training accuracy significantly but should enhance generalizability, since human speech does usually not exceed this frequency. `voxforge.org` provides files bundled into archives containing audio recorded by a single speaker. All files from

one archive were either placed in validation or training. The problem of the network recognizing speakers however could only be reduced not eliminated, since most speakers upload multiple archives and a vast amount of audio is uploaded anonymously.
The confusion matrix resulting from the classification of the validation set is shown in Fig. 4. An entry in the confusion matrix $C_{i,j}$ shows how often language i is classified as language j. Notably English is classified rather poorly here, which was not the case for the networks trained on the data from Topcoder, where it was a lot harder to distinguish French, Spanish and Italian.

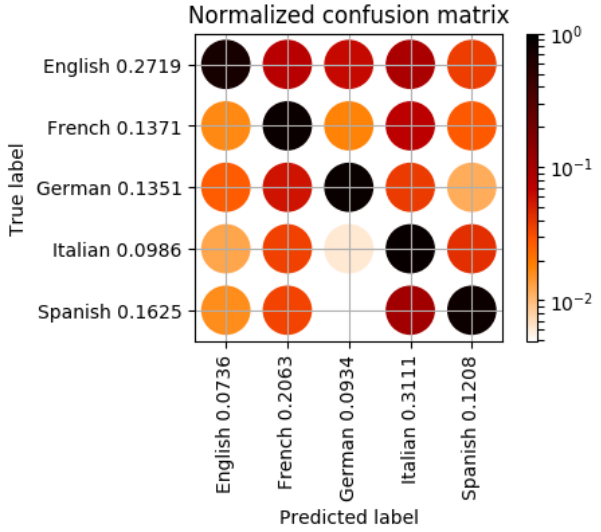

FIG. 4. *Confusion Matrix for the validation of the network trained on the dataset consisting of audio files downloaded from* `voxforge.org`. Numbers after the labels give the sum of the corresponding row or column without the diagonal entry. This number gives an indication, how often a language is misclassified as any other one for the rows. For the columns it indicates, how often other languages are wrongfully classified as the language given. A confusion matrix of the network trained on 176 languages can be found in supplemental material on GitHub [10].

To further analyze the correlation between the classified languages, the correlation matrix can be computed using the prediction vectors from audio files in the validation set:

$$\mathrm{corr}(x,y) = \frac{\mathrm{cov}(x,y)}{\sigma_x \sigma_y} = \frac{\mathrm{E}([x-\mu_x][y-\mu_y])}{\sigma_x \sigma_y},$$

where $\mathrm{cov}(x,y)$ is the covariance, μ_x, μ_y the expectation values of prediction distributions of languages x and y, and E the expected (mean) value of its argument. Applying a threshold, i.e. discarding specific correlation matrix elements, one obtains an adjacency matrix, which can be converted to an undirected graph to identify language clustering and connectivity. For the large dataset from Topcoder with 176 languages, a selected cluster for languages with positive correlations is shown in Fig. 5.

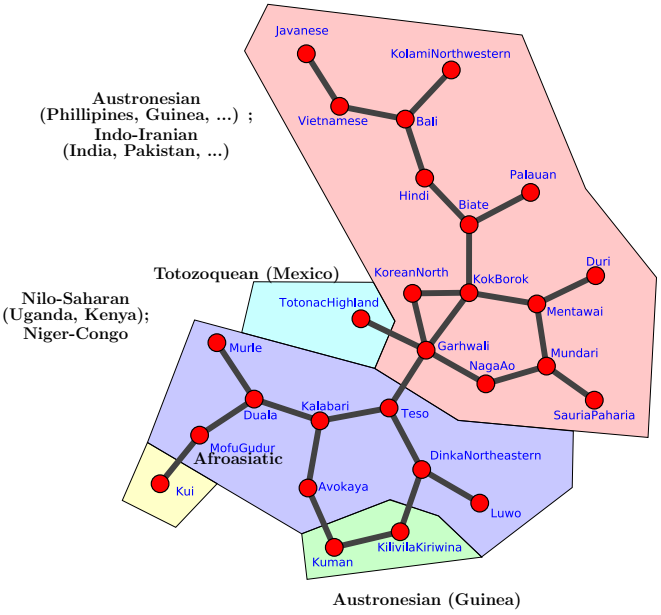

FIG. 5. *Positive correlation connections for a selected language cluster of the dataset with 176 classes. For this cluster we observe interconnections between Austronesian and Indo-Iranian languages (red) and Nilo-Saharan and Niger-Congo languages (blue). Some languages however, e.g. Totonac (turquoise), show correlations which disagree with modern linguistic findings. This cluster shows subclusters of languages corresponding to similar language families whose interconnections seem to be questionable (e.g. Teso and Garhwali). This might be due to additional arbitrary connections due to speaker voice similarity.*

DISCUSSION

The fact that for the data from `voxforge.org`, English often is mistaken for another language (indicated by the number in the first row), while other languages are rarely misclassified as English could lead to the conclusion that English was underrepresented in the training data, while in fact it was actually slightly overrepresented. However the Data uploaded by English speaking users may be less diverse or of poorer quality. Other than that, there are no really obviously outstanding properties of the confusion matrix in Fig. 4. Observations as that expect for English, other languages are rarely misclassified to be German is not surprising, due to the fact that English and German are germanic languages, while the others are romance (italic).

It is at first sight surprising that positive correlations between languages occur, which is not the case for the datasets containing only 5 languages. For a perfect network, the languages should all be anti-correleted, but in

this case, it seems intuitive that for a multiclass problem, some classes that are intrinsically correlated show positive correlations in the class predictions. In Fig. 5, we indeed observe positive correlations for familiar languages that share geographical origins and territories. However, some observed connections are obviously not in accordance with etymological linguistics. This could also be traced back to the strong speaker sensitivity of the network, yielding to a language correlation whose speakers have similar voice characteristics. Nonetheless, language similarities can be observed despite the adoption of the network to speakers voices, indicating that also important language-specific features were learned during the training.

OUTLOOK

This work clearly shows that the selection of audio data is a key requirement for a Spoken language recognition using convolutional networks. The audio fragments need to be diverse in recording settings, and most import speaker voices. Also, the dataset need to be balanced to prevent overfitting in favor of a overrepresented language. Thus, a larger and more reliable dataset of audio files would be of high importance to further increase the performance of spoken language classification. In this work we have dealt with convolutional networks only, but recurrent networks and combinations of recurrent and convolutional networks – particularly using GRU units – have been applied to speech recognition tasks with great success. [11] The architecture of our network could be more finetuned to recognize language characteristic patterns in the spectrograms, for instance by adjusting the kernel-size in the convolutional layers which were chosen to be of quadratic shape in both time and frequency dimension. Additional augmentation could have been used, for example to warp not only the frequency axis but also the time axis to mimic different speech pace. We were note able, to analyze the impact of various hyperparameters on the performance of the network in full detail, but tried to vary learning rate, dropout in the last layer, batch size, and steps per epoch to achieve fast convergence during the training. Thus, hyperparameters such as the choice of the optimizer, the kernel size, the number of channels in each layer or the total number of layers might not have been chosen optimally for training the dataset. Overall, this work shows that spoken language recognition is to no extent a trivial task and involves a careful selection of the dataset as well as a fine-tuned deep learning network that adjusted whose layers and hyperparameters are adjusted to the data. We have shown that convolutional networks are in principal a suitable choice for classifying spoken languages, although one has to deal carefully with speaker voice recognition.

[1] J. Benesty, M. M. Sondhi, and Y. Huang, *Springer handbook of speech processing* (Springer Science & Business Media, 2007).

[2] "Detecting Languages: Google Cloud Translation API Documentation: https://cloud.google.com/translate/docs/detecting-language, last viewed: 2017-07-18,".

[3] "TopCoder: Problem: SpokenLanguages (5 languages) https://community.topcoder.com/ longcontest/?module=viewproblemstatement&rd=16498&pm=13845, last viewed: 2017-08-08," (2017).

[4] "TopCoder: Problem: SpokenLanguages2 (176 languages) https://community.topcoder.com/ longcontest/?module=viewproblemstatement&rd=16555&compid =49304, last viewd: 2017-08-08," (2017).

[5] "VoxForge: voxforge.org, last viewd: 2017-08-08," (2017).

[6] L. Lee and R. Rose, "A frequency warping approach to speaker normalization," (1998).

[7] N. Jaitly and G. E. Hinton, in *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language* (2013) pp. 625–660.

[8] "Github: https://yerevann.github.io/2016/06/26/combining-cnn-and-rnn-for-spoken-language-identification/, last viewed: 2017-08-13," (2016).

[9] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," (2015).

[10] "Zusatzmaterial auf Github: https://github.com/amateurdeeplearners/language_recognition, last viewed: 2017-08-14," (2017).

[11] B. Chigier, "Automatic speech recognition," (1997), uS Patent 5,638,487.