Deep learning for spoken language identification

Grégoire Montavon
Machine Learning Group
Berlin Institute of Technology
Germany
gmontavon@cs.tu-berlin.de

Abstract Empirical results have shown that many spoken language identification systems based on hand-coded features perform poorly on small speech samples where a human would be successful. A hypothesis for this low performance is that the set of extracted features is insufficient. A deep architecture that learns features automatically is implemented and evaluated on several datasets.

1 Introduction

Spoken language identification is the problem of mapping continuous speech to the language it corresponds to. Applications of spoken language identification include front-ends for multilingual speech recognition systems, web information retrieval, automatic customer routing in call centers or monitoring.

Empirical results have shown that many systems based on the manual extraction of acoustic, prosodic, phonotactic or lexical features have significantly lower performance on small speech samples than on large speech samples [3, 4], while a human would still be successful. A hypothesis for this low performance is that the set of extracted features is insufficient [5].

Deep learning potentially addresses this issue by exploring the space of features automatically, bypassing the traditional phoneme recognition layer and learning instead purely discriminative features. A deep architecture is implemented and evaluated on several datasets.

2 Design and implementation

We train and evaluate our architecture on two datasets:

VoxForge This dataset consists of multilingual speech samples available on the VoxForge [9] website. This dataset contains 5 seconds speech samples associated with different metadata including the language of the sample. Given that speech samples are recorded by users with their own microphones, quality varies significantly between different samples. This dataset contains 25420 English samples, 4021 French samples and 2963 German samples.

RadioStream This dataset consists of samples ripped from several web radios. It has the advantage of containing a virtually infinite number of samples that are moreover of excellent quality. However, some samples are outliers, for example, music sequences or interviews in foreign languages. It means that the classification error is lower bounded by some constant $e \simeq 5\%$ also known as the Bayesian rate. A possible workaround consists of removing outliers manually from the test set, however, we don't use it because in certain cases the definition of "outlier" can be ambiguous. We use the following web radios:

English	KCRW, Newstalk, KALW
French	BFM, RFI, RTL, France Info
German	B5 Aktuell, B5 Plus, Deutsche Welle, NDR Info, HR Info

Figure 1: Spectrograms corresponding to a sample from the *VoxForge* dataset (left) and from the *RadioStream* dataset (right). Spectrograms showed here are truncated to 2.25 seconds (270 pixels) instead of 5 seconds (600 pixels). Spectrograms encode speech with 39 mel-frequencies between 0 and 5 kHz. Quality of spectrograms varies depending on the microphone, the voice of the speaker and the environmental noise.

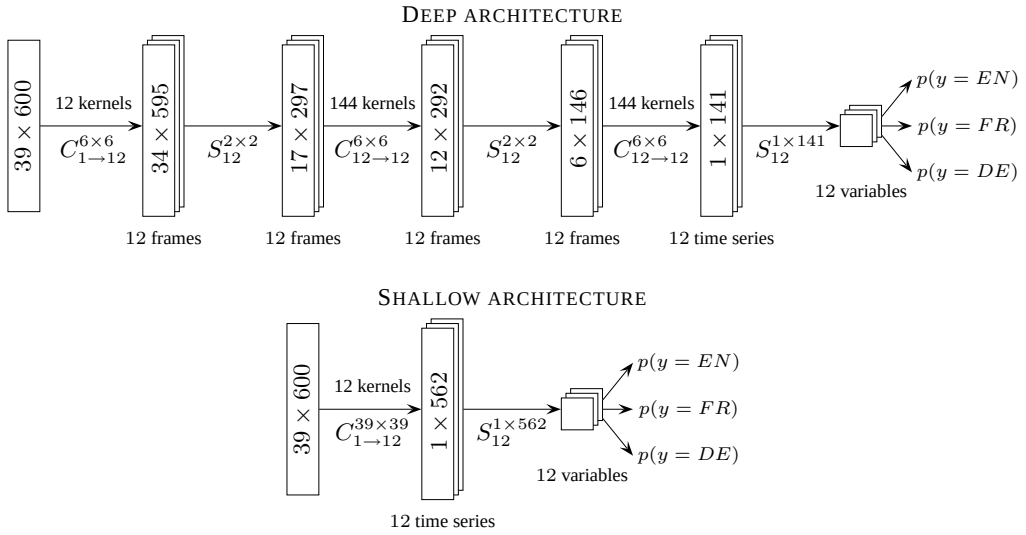

Figure 2: Deep and shallow CNN-TDNN architectures. A convolutional layer $C^{k\times l}_{m\to n}$ computes $m{\cdot}n$ convolutions between m input frames and n output frames with convolution kernels of size $k \times l$ and applies element-wise the nonlinearity $\max(\min(x, 1), -1)$ to the output. A subsampling layer $S^{k\times l}_{m}$ subsamples m input frames by a factor $k \times l$. The TDNN is implemented by the uppermost subsampling layer.

The classification problem consists of determining whether speech samples are English, French or German. These languages are chosen because both datasets contain a sufficient number of samples for each of them. We train and evaluate the classifier on balanced classes (33% English samples, 33% French samples and 33% German samples). Each sample corresponds to a speech signal of 5 seconds.

For each speech signal, a spectrogram of 39×600 pixels is constructed where the y-axis represents 39 mel-frequencies between 0 and 5 kHz and the x-axis represents 600 observed times spaced by 8.33 milliseconds. Each frequency of the spectrogram is captured using a Hann window. Examples of spectrograms are given in figure 1. The range 0–5 kHz is chosen because most of the spectral power of speech falls into that range.

The classifier maps spectrograms into languages and is implemented as a time-delay neural network (TDNN) with two-dimensional convolutional layers as feature extractors. Our implementation of the TDNN performs a simple summation on the outputs of the convolutional layers. The architecture is implemented with the Torch5 [8] machine learning library and is presented in figure 2.

Using a TDNN is motivated by good results obtained for speech recognition [2, 7]. Using convolutional layers as feature extractors is motivated by good results obtained by convolution-based architectures such as convolutional neural networks (CNN) for various visual perception tasks such as handwriting digit recognition [6]. The classifier is trained with a stochastic gradient descent [1].

VoxForge

Deep architecture — *Known speakers*

	EN	FR	DE
EN	33.4	0.6	0.3
FR	1.9	30.8	0.6
DE	4.5	0.9	26.9

Accuracy $= 91.2\%$

Deep architecture — *New speakers*

	EN	FR	DE
EN	33.0	0.8	1.4
FR	2.8	27.4	1.4
DE	12.0	1.6	19.6

Accuracy $= 80.1\%$

Shallow architecture — *New speakers*

	EN	FR	DE
EN	28.3	1.2	4.4
FR	3.7	26.7	2.5
DE	10.5	3.1	19.7

Accuracy $= 74.6\%$

RadioStream

Deep architecture — *Known radios*

	EN	FR	DE
EN	28.7	1.5	3.6
FR	1.3	28.6	2.1
DE	2.3	1.5	30.4

Accuracy $= 87.7\%$

Deep architecture — *New radios*

	EN	FR	DE
EN	28.0	1.7	5.5
FR	1.4	27.7	2.5
DE	2.9	2.5	27.8

Accuracy $= 83.5\%$

Shallow architecture — *New radios*

	EN	FR	DE
EN	22.1	2.7	7.9
FR	4.0	23.3	5.4
DE	4.8	2.0	27.6

Accuracy $= 73.1\%$

Figure 3: Performance of the classifier on 5 seconds speech samples. Rows of the confusion matrices represent the true label and columns represent the prediction of the classifier. Accuracy is computed as the trace of the confusion matrix.

3 Results and analysis

The performance of the deep architecture presented in figure 2 is evaluated on *VoxForge* and *RadioStream* datasets presented in section 2 in two different settings:

1. Classification for *known speakers and known radios*: speech samples are randomly assigned to the training and test set with a respective probability of 0.5 and 0.5.
2. Classification for *new speakers and new radios*: on *VoxForge*, speech samples coming from speakers with initials [A-P] are assigned to the training set and speakers with initials [Q-Z] to the test set. On *RadioStream*, speech samples coming from KALW, France Info and HR Info are assigned to the test set and the remaining ones to the training set.

We compare the deep architecture with the shallow architecture also presented in figure 2. Choosing convolution kernels of size 39×39 for the shallow architecture is motivated by the fact that the subsequent numbers of weights for both architectures have the same order of magnitude ($\sim 10^4$ weights) and that both architectures are then able to model 39 pixels of time dependence. Time dependence is measured as the time interval occupied by the subset of input nodes connected to a single hidden node located just before the uppermost subsampling layer. The deep architecture has $2.8 \cdot 10^7$ neural connections against 10^7 for the shallow architecture and takes consequently 2.8 times longer to train. We train the deep architecture for $0.75 \cdot 10^6$ iterations and the shallow architecture for $2.8 \cdot (0.75 \cdot 10^6) = 2.1 \cdot 10^6$ iterations so that both architectures benefit from the same amount of computation time. Controlling the number of parameters, the amount of time dependence and the number of iterations allows to effectively measure the influence of depth on language identification. Results are presented in figure 3. We observe the following:

1. The deep architecture is 5–10% more accurate than its shallow counterpart. Translation invariances are not directly encoded by the structure of the shallow architecture and must therefore be inferred from the data, slowing down the convergence time and leading to poor generalization when the data is limited.
2. The neural network builds better discriminative features between French and non-French samples than between English and German samples. A possible explanation is that German and English are perceptually similar due to their common West-Germanic ancestor. It shows that the overall accuracy of a system can vary considerably depending on the selected subset of languages to identify.
3. On the *VoxForge* dataset, samples from new German speakers are often misclassified. It seems that the low number of German samples or the low number of German speakers

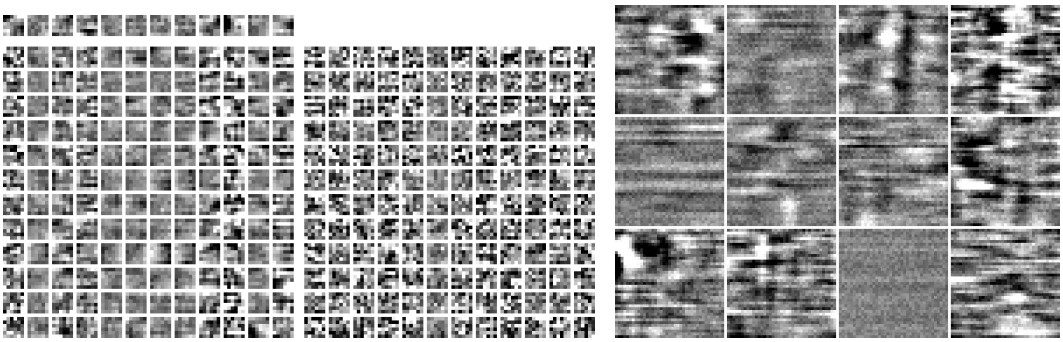

Figure 4: Convolution kernels obtained on the *VoxForge* dataset. On the left: the $12 + 144 + 144$ convolution kernels of size 6×6 of the deep architecture. On the right: the 12 convolution kernels of size 39×39 of the shallow architecture. In both cases, not all convolution kernels are used, which means that the capacity of the neural network is not fully used and that the performance bottleneck is not the number of frames in the hidden layers but rather the distance between train and test data, the presence of local minima in the loss function or the structure of the neural network.

prevents the classifier from creating good "German" features. The sensitivity to the number of samples or speakers is an argument for collecting more samples from more speakers.

4. Samples from known speakers are not classified perfectly. While figure 4 suggests that the number of frames in each hidden layer is sufficient, 39 pixels of time dependence might not be sufficient to create lexical or syntactic features. Solutions to increase time dependence are (1) to increase the size of the convolution kernels and control the subsequent risk of overfitting by using more samples or (2) to replace the last averaging module by a hierarchy of convolutional layers and, if necessary, handle the subsequent depth increase by training the new architecture greedily layer-wise.

4 Conclusion

A deep architecture for spoken language identification is presented and evaluated. Results show that it can identify three different languages with 83.5% accuracy on 5 seconds speech samples coming from radio streams and with 80.1% accuracy on 5 seconds speech samples coming from *VoxForge*. The deep architecture improves accuracy by 5–10% compared to its shallow counterpart. It indicates that depth is important to encode invariances required to learn fast and generalize well on new data. While we emphasize the superiority of deep architectures over shallow ones for this problem, it remains to determine how deep learning compares to techniques based on hand-coded features. We suggest that accuracy can be improved by (1) collecting more samples from more speakers and (2) extending time dependence in order to learn higher level language features.

References

[1] L. Bottou, Stochastic Gradient Learning in Neural Networks, 1991

[2] L. Bottou, Une Approche théorique de l'Apprentissage Connexionniste: Applications à la Reconnaissance de la Parole, 1991

[3] J. Hieronymous and S. Kadambe, Spoken Language Identification Using Large Vocabulary Speech Recognition, 1996

[4] R. Tong, B. Ma, D. Zhu, H. Li and E.-S. Chng, Integrating Acoustic, Prosodic and Phonotactic Features for Spoken Language Identification, 2006

[5] R. Cole, Survey of the State of the Art in Human Language Technology, 1997

[6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition, 1998

[7] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano and K. J. Lang, Phoneme recognition using time-delay neural networks, 2002

[8] R. Collobert, Torch5, `www.torch5.sf.net`

[9] VoxForge, Free Speech Recognition, `www.voxforge.org`