Automatic Segmentation and Deep Learning of Bird Sounds

Hendrik Vincent Koops(✉), Jan van Balen, and Frans Wiering

Department of Information and Computing Sciences,
Utrecht University, Utrecht, The Netherlands
{h.v.koops,j.m.h.vanbalen,f.wiering}@uu.nl

Abstract. We present a study on automatic birdsong recognition with deep neural networks using the BIRDCLEF2014 dataset. Through deep learning, feature hierarchies are learned that represent the data on several levels of abstraction. Deep learning has been applied with success to problems in fields such as music information retrieval and image recognition, but its use in bioacoustics is rare. Therefore, we investigate the application of a common deep learning technique (deep neural networks) in a classification task using songs from Amazonian birds. We show that various deep neural networks are capable of outperforming other classification methods. Furthermore, we present an automatic segmentation algorithm that is capable of separating bird sounds from non-bird sounds.

Keywords: Deep learning · Feature learning · Bioacoustics · Segmentation

1 Introduction

Features are predominantly handcrafted in audio information retrieval research. For a successful translation from heuristics to algorithmic methods, a significant amount of domain- and engineering knowledge is needed. Creating features from heuristics depends on the assumption that the feature designer can know what a good representation of a signal must be to solve a problem. Feature design is thus constrained by what a designer can conceive and comprehend. Furthermore, manual optimization of handcrafted features is a slow and costly process.

A research area that tries to solve some of the aforementioned problems in feature design is called *deep learning*, in which multilayer architectures are used to learn *feature hierarchies*. The more abstract features that are higher up in the hierarchy are formed by the composition of less abstract features on lower levels. These multi-level representations allow a deep architecture to learn the complex functions that map the input (such as digital audio) to output (e.g. classes), without the need of dependence on handcrafted features [11].

Related Work. Feature learning has been succesfully applied in music information retrieval tasks such as musical genre [5], and emotion recognition [6]. Deng and Yu [7] argue that automatic learning of feature hierarchies, and high level

J. Mothe et al. (Eds.): CLEF 2015, LNCS 9283, pp. 261–267, 2015.
DOI: 10.1007/978-3-319-24027-5_26

features in particular, will become more important as the amount of data and range of machine learning applications continues to grow. Therefore, we investigate the application of deep neural networks (DNN) in classification of a large birdsong corpus. This paper extends previous work [12] with network strategies to prevent overfitting.

Contribution. The contribution of this paper is threefold. First, it provides the first results of applying DNN in classification of bird songs. Secondly, this paper provides a novel algorithm to automatically segment noisy bird sounds into bird- and non-bird sounds. Thirdly, this paper sets a baseline towards the application of state of the art feature learning algorithms in bioacoustics.

The remainder of this paper is structured as follows. Section 2 details bird sound segmentation. Section 3 describes classification using DNN. Section 4 presents classification results. Concluding remarks can be found in Section 5.

2 Automatic Segmentation of Noisy Bird Sounds

Often, a substantial part of a birdsong recording contains background noise. Therefore, we create a segmentation algorithm that is based on the assumption that the loudest parts of a signal are the most relevant. The algorithm consists of three parts: 1: decimating and filtering, 2: segmenting and 3: clustering.

Decimating and Filtering. Decimation of a signal is common practice in speech recognition, as it reduces the amount of information by removing the top part of the spectrum that we know cannot hold the most important information. The spectrum energy in song birds is typically concentrated on a very narrow area in the range of 1 to 6 kHz [2]. Therefore, we down-sample birdsong recordings by a factor 4, resulting in a maximum signal frequency of 5.5125 KHz for signals with a sample rate of 44100 Hz. Although some bird song frequencies could exist beyond this limit, this is never the loudest frequency. After decimation, the signal is passed through a 10^{th} order high pass filter with a passband frequency of 1kHz and a stop band attenuation of 80 dB to filter unwanted low frequency noise. Finally, the signal is passed through another 10^{th} order high pass filter to account for sounds that occur below the bird sound in the spectrogram. This filter varies its passband frequency to $0.6 * f_m$ per signal, where f_m is the the maximum value of the signal's spectrogram.

Segmentation. We segment a recording into bird sounds and non-bird sounds by finding the maximum sections of a spectrogram using a an energy-based algorithm somewhat similar to [3]. In the spectrogram of a signal f, the peak of f at time t_n is found. From this peak, a left and right wise trace is performed until the value at the trace position falls below a threshold φ dB, which indicates the boundary of a segment. Tracing is repeated until no untraced peak above the threshold is found, resulting in n segments per recording. In a manual inspection, $\varphi = 17$ was found to create the best segments.

Clustering. An unwanted artifact of the aforementioned segmentation is the creation of a large number of small segments of only a few milliseconds (ms) in length. Bird songs are better described at a higher temporal level, which is richer in information. Therefore, we merge segments by analyzing the distances between sections and combining subsequent segments with distances smaller than m ms. Segmentation is evaluated in an experiment where handcrafted segments are compared to automatically generated segments [10]. $m = 800$ms was found to create segments that closely match human annotations.

3 Deep Neural Network Classification

Figure 1 shows an example of a DNN. We use a multilayer neural network that is fully connected between layers, also called a *deep belief network* [13]. The networks are initialized using a greedy layer-wise unsupervised pre-training phase, thereby initializing the network closer to a good solution than random initialization. This avoids local minima when using supervised gradient descent [14]. After pre-training, gradient descent learning is used to train and fine-tune the networks. To explore the effects of hidden layer size on classification, we create two types of networks: one in which the hidden layer size is smaller or equal to the input layer, and one where the hidden layer is larger than the input layer. The classification layer is always of a fixed size in every network, corresponding to the number of species classes in a dataset. We also experiment with *dropout*, [1] to avoid overfitting. In dropout, half of the nodes of the hidden layers of the neural networks are randomly omitted on each training case, by setting their value to 0 with a probability of 0.5 on each training iteration. This prevents complex co-adaption in which hidden layer activation is only helpful in the context of other specific hidden layer activation.

Batch Optimization. To update the parameters of the networks, we use *mini-batch optimization*. With this method, the parameters of the networks are updated using the summed gradients measured on a rotating subset of size n

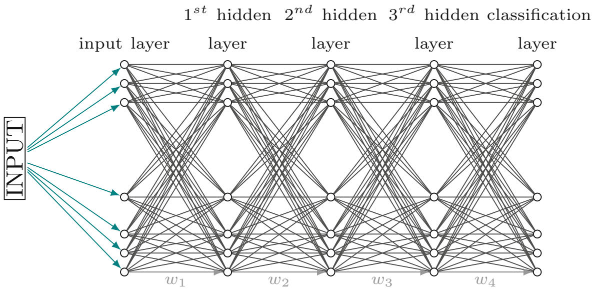

Fig. 1. Example of a Deep Neural Network with three hidden layers.

Table 1. Contents of three different MFCC datasets.

	Mean	Variance	Speed	Acceleration	Means of three subsections
$\mathcal{D}_{48}$	✓				
$\mathcal{D}_{96}$	✓	✓			
$\mathcal{D}_{240}$	✓	✓	✓	✓	✓

of a training set. During early testing and implementation, it was found that a batch size of $n = 250$ returned favorable results.

Voting. The input to the networks are segments of recordings, and the networks therefore perform *segment based* classification. To be able to classify individual *recordings*, we use voting to combine the classifications of segments of a recording. We use an approach that uses the classification layer activations as probabilities, thereby taking advantage of the network's classification uncertainty. For each segment in a recording, a vector is created in which the classes are added proportionally to their activation in the classification layer (e.g. a class with activation 0.1 is added 1 time, 0.4 is added 4 times, etc). Finally, the mode over the vectors of all segments is chosen as the class for the recording.

4 Results

The BirdCLEF2014 ($\mathcal{BC}_{14}$) [4] dataset is used for evaluation. $\mathcal{BC}_{14}$ was released for the 2014 BirdCLEF task and contains around 14000 audio recordings of 501 South American bird species. The segments created by our algorithm are used to select Mel-Frequency Cepstrum Coefficients (MFCC) features from a MFCC dataset that was included in the $\mathcal{BC}_{14}$. MFCC are coefficients that together represent the power spectrum of a sound on a scale that tries to mimic human perception of pitch. Originally designed for speech processing applications, they have since been successfully used in bioacoustics research [8,9].

Using the segments, we create three MFCC datasets ($\mathcal{D}_{48}$, $\mathcal{D}_{96}$ and $\mathcal{D}_{240}$), of which the contents is listed in Table 1. Each dataset contains 46799 segments (4.83 segments per recording). The datasets are shuffled per recording and divided into a 80% train and 20% test set, and together with their classes used as input for several DNN. The classification results of several network topologies are presented in Table 2. Network topologies are notated as a series of layer sizes. "48-(40×2)-501" denotes a DNN with 48 input nodes, two layers of hidden nodes with 40 nodes and an output layer of 501 nodes. 48-networks are trained and tested with $\mathcal{D}_{48}$, 96-networks with $\mathcal{D}_{96}$ and 240-networks with $\mathcal{D}_{240}$.

We find that classification accuracy increases with the size of the network, except for the 48-networks. In the 96-networks, a big jump in accuracy is observed with regard to the 48-networks, to around 10% in the 96-networks without dropout and around 6% accuracy with dropout. Training accuracy is high in the 96-networks without dropout, while the testing accuracy is low. The training and testing accuracy of the 96-networks with dropout are lower, but closer together, showing that dropout was effective in preventing overfitting.

Table 2. Train and test results of various network topologies. In the columns on the right *"+d"* denotes drop-out, *"+v"* denotes voting. Best performance is highlighted.

Network topology	Train (segments)	Test (segments)	Test+v (recordings)	Network topology	Train (segments)	Test (segments)	Test+v (recordings)
48-(40×2)-501	45.6%	0.5%	0.32%	48-(40×3)-501 +d	4%	0.27%	0.13%
48-(48×2)-501	52%	0.34%	0.27%	48-(48×3)-501 +d	3%	0.30%	0.11%
96-(64×2)-501	77.6%	9.34%	10.05%	96-(64×3)-501 +d	10%	5.97%	5.05%
96-(84×2)-501	85%	10.55%	11.35%	96-(84×3)-501 +d	17%	8.10%	7.40%
240-(128×2)-501	15.0%	10.03%	11.25%	240-(128×3)-501 +d	22%	9.51%	9.11%
240-(350×2)-501	10.0%	11.03%	12.08%	240-(350×3)-501 +d	51%	13.83%	13.23%

Table 3. Results of two non-neural network classifiers. Best performance is highlighted.

Classifier	Dataset	Train accuracy	Test accuracy
Rotation Forest (RF)	$\mathcal{D}_{48}$	99.979%	0.24%
Rotation Forest (RF)	$\mathcal{D}_{96}$	26.686%	8.99%
Rotation Forest (RF)	$\mathcal{D}_{240}$	100%	8.25%
Support Vector Machines (SVM)	$\mathcal{D}_{48}$	7.086%	1.03%
Support Vector Machines (SVM)	$\mathcal{D}_{96}$	29.75%	10.17%
Support Vector Machines (SVM)	$\mathcal{D}_{240}$	29.64%	10.06%

$\mathcal{D}_{240}$ supplements the $\mathcal{D}_{96}$ with the means of three equal subsections of a segment. This extra information improves only a little bit in the 240-networks with a hidden layer of size 350 without dropout. In the networks without dropout, the 240-network with hidden layer size 128 performs worse than the 96-network with hidden layer size 84, but better than the 96-network with hidden layer size 64. The 240-networks outperform other networks with dropout. The difference between test and train accuracy in the dropout networks increases with the size of the networks, but this is not observed in the networks without dropout. Overall, the largest network (240-(350×3)-501) with dropout the best classifier.

Other Classification Methods. Table 3 shows the classification accuracies of the $\mathcal{D}_{48}$, $\mathcal{D}_{96}$ and $\mathcal{D}_{240}$ on two non-neural network classifiers. Again it is found that using only the mean of the MFCC in a segment ($\mathcal{D}_{48}$) produces classification accuracies close to random classification. This holds for both Rotation Forest (RF) and Support Vector Machines SVM, with the former accurately classifying only 0.235% of the examples and the latter 1.026% of the examples. RF performs below random classification and SVM above the random baseline of 0.3%. A big jump in classification accuracy with both methods is observed when adding the variance ($\mathcal{D}_{96}$). Additionally adding the means of three subsections by using the $\mathcal{D}_{240}$-set decreases the classification accuracy of RF, compared to $\mathcal{D}_{96}$, but outperforms the $\mathcal{D}_{48}$. Overall, SVM produces best result for this task (10.17%).

5 Discussion and Conclusions

The results from the Table 2 and 3 show that DNN are capable of outperforming RF and SVM, when taking into account all datasets. The BirdClef committee reported that the random baseline in this task was 0.3%, which is comparable to the the smallest DNN used in the experiments. Using $\mathcal{D}_{96}$, DNN outperform the other tested classification methods. The best 96-network (96-(84×2)-501) outperforms SVM by 1.2% and RF by 2.4%. Comparing the results of the 96-networks with those of the 48-networks shows that important information of birdsong is contained in the variance of the MFCC, indicating that how coefficients vary over time is important in discriminating species. The best results are obtained using the 240-set on DNN with and without dropout. Overall, these results show that adding time-varying information is vital to the classification of birdsongs using MFCC. Furthermore, is is shown that DNN are capable of outperforming SVM and RF on several MFCC datasets. The results of this paper show that deep learning is valuable to bioacoustics research and bird song recognition.

Acknowledgments. Hendrik Vincent Koops is supported by the Netherlands Organization for Scientific Research, through the NWO-VIDI-grant 276-35-001 to Anja Volk. Frans Wiering is supported by the FES project COMMIT/. Jan Van Balen is supported by Cogitch (NWO CATCH project 640.005.004).

References

1. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. CoRR abs/1207.0580 (2012)
2. Harma, A.: Automatic identification of bird species based on sinusoidal modeling of syllables. In: Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003), vol. 5 (2013)
3. Somervuo, P., Harma, A.: Bird song recognition based on syllable pair histograms. In: 2004 Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2004), vol. 5, p. V–825. IEEE (2004)
4. Goëau, H., Glotin, H., Vellinga, W.-P., Rauber, A.: LifeCLEF bird identification task 2014. In: CLEF Working Notes 2014 (2014)
5. Hamel, P., Eck, D.: Learning features from music audio with deep belief networks. In: Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010), pp. 339–344. Utrecht, The Netherlands (2010)
6. Schmidt, E., Scott, J., Kim, Y.: Feature learning in dynamic environments: modeling the acoustic structure of musical emotion. In: Proceedings of the 13th International Society for Music Information Retrieval Conference, Porto, Portugal, October 8–12, 2012
7. Deng, L., Yu, D.: Deep learning: Methods and applications. Technical Report MSR-TR-2014-21, Microsoft, January 2014
8. Lee, C.-H., Lee, Y.-K., Huang, R.-Z.: Automatic recognition of bird songs using cepstral coefficients. Journal of Information Technology and Applications **1**(1), 17–23 (2006)
9. Chou, C.-H., Ko, H.-Y.: Automatic birdsong recognition with MFCC based syllable feature extraction. In: Hsu, C.-H., Yang, L.T., Ma, J., Zhu, C. (eds.) Ubiquitous Intelligence and Computing. LNCS, vol. 6905, pp. 185–196. Springer, Heidelberg (2011)
10. Koops, H.V.: A Deep Neural Network Approach to Automatic Birdsong Recognition. Master's Thesis, Utrecht University (2014)

11. Bengio, Y.: Learning deep architectures for AI. In: Foundations and trends® in Machine Learning, pp. 1–127. Now Publishers Inc. (2009)
12. Koops, H.V., Van Balen, J., Wiering, F.: A deep neural network approach to the LifeCLEF 2014 bird task. In: CLEF 2014 Working Notes, vol. 1180, pp. 634–642 (2014)
13. Hinton, G.E., Osindero, S., Teh, Y.-W.: A fast learning algorithm for deep belief nets. Neural Computation **18**(7), 1527–1554 (2006). MIT Press
14. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. In: Advances in Neural Information Processing Systems 19, pp. 153–160. MIT Press (2007)