EXPLORING LARGE AUDIO MODEL REDUNDANCY IN BIRD SOUND CLASSIFICATION

Marco Andreis (s243116), Rasmus Porsgaard (s203953), Csilla Duray (s233903)

Technical University of Denmark

ABSTRACT

Our project focused on the exploration of large audio model pruning, investigating the redundancy inside their layers in a specific bioacoustic task. Bioacoustics is a rapidly evolving field since the application of machine learning models for monitoring biodiversity. Transformer models that are fine-tuned for downstream tasks, such as the classification of various species based on audio data, may be pruned for optimised inference time, while retaining their performance.

Here, we demonstrate the pruning process of a wav2vec2 model fine-tuned for a bird sound classification task. We rely on various similarity metrics and convexity measures to do so, and compare our final model performance with baseline models such as a simple CNN and the feature extractor.

Index Terms— wav2vec, bioacoustics, speech representation learning, model pruning

1. INTRODUCTION

The field of bioacoustics — studying non-human species through sound — has advanced significantly with the development of increasingly complex machine learning models. Training models on animal sounds has proven valuable for tasks such as species identification and biodiversity monitoring [1, 2]. Birds, in particular, serve as an excellent focus for bioacoustic studies due to their extensive reliance on sound for communication. Their vocalizations, which can include mating calls, warning signals, and territorial markers, showcase an incredible diversity enabled by their unique anatomy [3]. Moreover, bird vocal behaviour is often indicative of ecological richness, making them key indicators of biodiversity.

While advanced machine learning architectures like transformer models have shown impressive performance in bioacoustic tasks, their complexity does not always result in better outcomes [4]. This project focuses on optimising a fine-tuned model for bird sound classification by demonstrating a practical approach to algorithm pruning.

Our goal is to develop a model architecture selection methodology that minimises model complexity without compromising performance. By fine-tuning and pruning the wav2vec2 model, we leverage similarity and convexity metrics to identify and remove redundant layers, ensuring the pruned model retains at least 95% of the accuracy of the fully fine-tuned version.

2. DATA AND MODEL ARCHITECTURES

We used a publicly available dataset of bird songs for developing a species classification model. To explore the relationship between model complexity and performance, we experimented with two different base models: the transformer-based wav2vec2-base-960h and a simpler convolutional neural network (CNN). Our choices of dataset, preprocessing steps and model were inspired by Hugging Face user dima806's work [5].

2.1. The Bird Song Dataset

We used 2,161 audio files of bird songs for our model training that were linked to a dataset containing their metadata, including their species. The data was scraped from Xeno-canto, a website that collects and shares wildlife sounds with the public [6, 7]. The preprocessing of the dataset involved filtering out the species that were recorded less than 25 times, which resulted in a dataset of 1,490 recordings and 50 of the most common species. The audio files were segmented into 10-second clips and converted into one-dimensional audio representations sampled at 16 kHz.

2.2. Wav2vec2 Model

The wav2vec2 model is a state-of-the-art transformer-based audio model, known for its success in various speech-related tasks, such as automatic speech recognition, speaker identification, and emotion detection. Originally introduced by Baevski et al. [8], wav2vec2 was pre-trained on the Librispeech dataset and achieved word error rates (WER) of 1.8% and 3.3% for clean and noisy speech transcription, respectively.

The model processes audio sampled at 16 kHz, analysing up to 25 ms of raw sound waves per frame. The wav2vec2 architecture comprises a feature encoder, transformer layers, and a linear layer for classification (*Figure 1*). The feature encoder is a stack of seven convolutional layers, each with 512 channels, to extract continuous feature representations from the

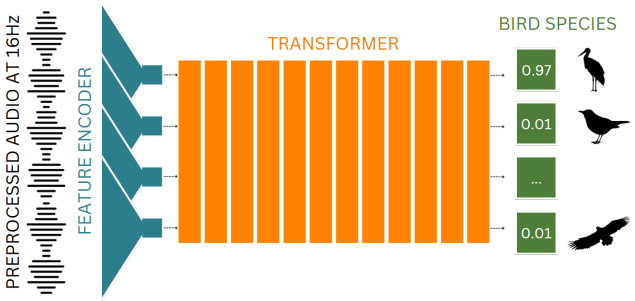


Fig. 1. *Fully fine-tuned model architecture based on wav2vec2-960h-base*

raw audio input. The "base" wav2vec2 model has 12 transformer blocks of 8 attention heads in each. For classification, a Glorot-initialised linear layer is added as the final component.
These transformer layers use attention mechanisms and feed-forward networks to learn meaningful representations from audio data. Notably, wav2vec2 employs grouped convolution layers to derive relative positional embeddings, enhancing its capacity for sequence modeling. While originally designed for human speech, its ability to process high-resolution audio data (16 kHz) aligns well with bird sound analysis, making it a viable choice for our task.
The model's final layers include projection and classification components, enabling it to adapt its learned representations for species classification.

2.3. Convolutional Neural Network (CNN)

As a baseline, we trained a simpler convolutional neural network (CNN) with two convolutional layers for extracting local patterns from the audio data and two fully connected layers, to aggregate features and classify the input into one of the 50 species. This simpler model served as a comparison for assessing the trade-offs between model complexity and classification performance.

3. METHODS

3.1. Similarity

The ability of a neural network to perform classification depends on the possibility of separating examples belonging to different classes by building latent representations which become increasingly more isolated flowing through the layers. For this reason, an approach for investigating the presence of redundancies in a model architecture consists in extracting the high-dimensional representation space constructed by each layer and quantifying the difference between these. The case where two consecutive layers show a high degree of similarity is indicative of redundancy as the computation performed by the second layer does not aid in the separation of the examples.
To identify similarity across layers, we followed the approach applied by Dorszewski et. al. [9] based on three similarity metrics: cosine similarity, centered kernel alignment (CKA) and mutual k-nearest neighbours (mutual kNN).

Let $i \in \{1, \ldots, 12\}$ be index of transformer layers and $f^{(i)} : \mathbb{R}^D \to \mathbb{R}^d$ the network up that layer. Let $X \in \mathbb{R}^{n \times D}$ be the input data composed of n D-dimensional examples. We use the notation $A^{(i)} = f^{(i)}(X) \in \mathbb{R}^{n \times d}$ to indicate the representation of input data after the i-th layer.
The cosine similarity score between representation of X produced after the i-th and j-th layers is defined as:

$$S_{\cos}(i,j) = \frac{1}{n} \sum_{l=1}^{n} \frac{\left(A_{l,\cdot}^{(i)}\right)^{\top} A_{l,\cdot}^{(j)}}{\|A_{l,\cdot}^{(i)}\| \|A_{l,\cdot}^{(j)}\|} \tag{1}$$

The CKA score [10], computed using a linear kernel, is:

$$S_{\text{CKA}}(i,j) = \frac{\left\| \left(A^{(i)}\right)^{\top} A^{(j)} \right\|_F^2}{\left\| \left(A^{(i)}\right)^{\top} A^{(i)} \right\|_F \left\| \left(A^{(j)}\right)^{\top} A^{(j)} \right\|_F} \tag{2}$$

Lastly, we consider the mutual kNN score [11]:

$$S_{\text{kNN}}(i,j) = \frac{1}{n} \sum_{l=1}^{n} \left(\frac{1}{k} \left| \mathcal{N}_k \left(A_{l,\cdot}^{(i)}\right) \cap \mathcal{N}_k \left(A_{l,\cdot}^{(j)}\right) \right| \right) \tag{3}$$

3.2. Graph convexity

Graph convexity, defined as follows:

Let (V, E) be a graph and $A \subseteq V$. A is convex if for all pairs $x, y \in A$, there exists a shortest path $P = (x = v_0, v_1, v_2, \ldots, v_{n-1}, y = v_n)$ and $\forall i \in \{0, \ldots, n\} : v_i \in A$.

was introduced by Tětková et al.[12] as an indicator of performance in models trained for the classification of data of different nature (audio, text, images…) and was later adopted by Dorszewski et al. [4], who showed that convexity is a useful measure in in guiding pruning decisions in large audio models.
Construction of the graph representation of the latent space of a given layer is achieved by extracting all the data points (vertices) and calculating the euclidean distance (edges weights) between each pair, of which only the 10 nearest neighbour are kept. The convexity score is assigned by finding the shortest path between all pairs of vertices belonging to the same class and taking the proportion of vertices on the path belonging to the same class as the endpoints (endpoints are not included in the proportion). The scores are averaged across all pairs and all classes to get a single value for the layer.

3.3. Pruning approach

In the transformer section of the network, the dimensions of each layer output are consistent, this characteristic of the architecture makes it extremely easy to perform pruning by feeding the output of the i-th layer to the $i + 2$-th layer, effectively removing the intermediate layer.
During the project, two pruning approaches were utilised. Post-training pruning consisted in finetuning the base wav2vec, performing analysis to identify blocks of layer with high similarity and then, through trial and error, using pruning to find the smallest model capable of maintaining the majority of the accuracy of the full-finetuned one. The first layer of the transformer section was always maintained as it is the one interfacing with the feature extractor, making it essential. The opposite was done in pre-training pruning, for which we backward pruned the base model one layer at a time, and then proceeded to the finetuning.

4. RESULTS

4.1. Pre-training pruning

Before finetuning, the base wav2vec2 achieves an accuracy of 3.31%, higher then the expected 2% baseline. Investigating the convexity profile (*Figure 2*) of the model, it can be seen that the initial layers seem capable of producing significantly isolated representations for the different classes, but the convexity decreases significantly proceeding in the network. This suggested that the full architecture could be not the most efficient, or at least not necessary, to the downstream classification task. For this reason we pruned the original model by considering an increasing number of layer and compared the performance achieved by each of them after finetuning (*Figure 3*). As indicated by the convexity, we observed how using only 5 layers was enough to reach a plateau in performance and, even though the peak in accuracy is obtained by using 7 layers, adding additional layers after the fifth does not provide significant improvements.

4.2. Post-training pruning

The finetuning of the full (12 blocks) wav2vec2 provided a model with 91% accuracy on the selected test set. The first step for pruning was to investigate the similarity profile of the network using the CKA, mutual kNN and cosine similarity metrics. Results of the analysis (*Figure 4*) highlight different characteristics: cosine similarity provides useful insight only about the presence of a similarity block containing layer 10-12. This consideration is in contrast with the mutual kNN matrix which shows no similarity between these layers, while it shows that each layer has some similarity with its surroundings. Lastly, the CKA indicates the presence of three similarity blocks consisting approximately in layers 1-4, 4-10, 11-12.

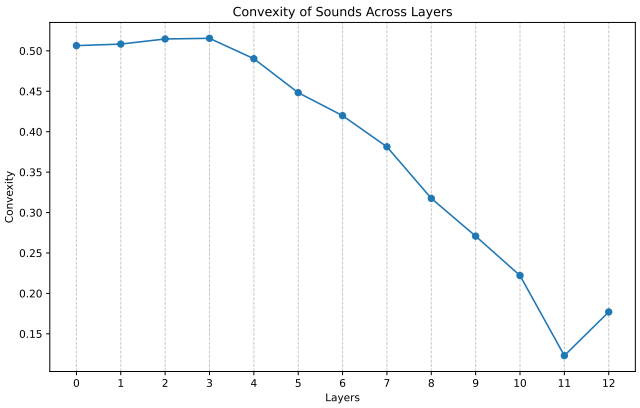


Fig. 2. *Performance achieved by increasingly larger networks.*

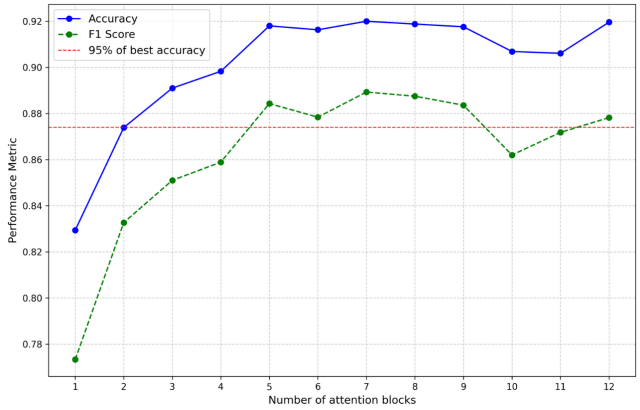


Fig. 3. *Performance achieved by increasingly larger networks.*

The pruning experiments (*Figure 5* shows the most relevant) started with forward, backward, middle and single layer pruning, allowing us to identify the impact of each layer on the overall classification ability of the network. The pruning of single layers revealed that the initial layers are extremely important for the performance of the network, this is shown by the drop in accuracy (10%) that is obtained by removing any of them, this is also confirmed by the results obtained with forward pruning. On the other side, backward showed that, as hypothesised from the CKA and cosine similarity matrix, the last 4 layers are redundant and it is possible to remove multiple losing 5-6% of accuracy. We identified redundancy, even though to a lesser extent, in layers from 5 to 8, with the latter being the least important. Following this reasoning we investigated, with a trial and error approach, which combination of layers provided the smallest network still maintaining more than 95% of the full network, which resulted in pruning layers 8-10-11 and additionally 4, if the accepted threshold is lowered to 90%.

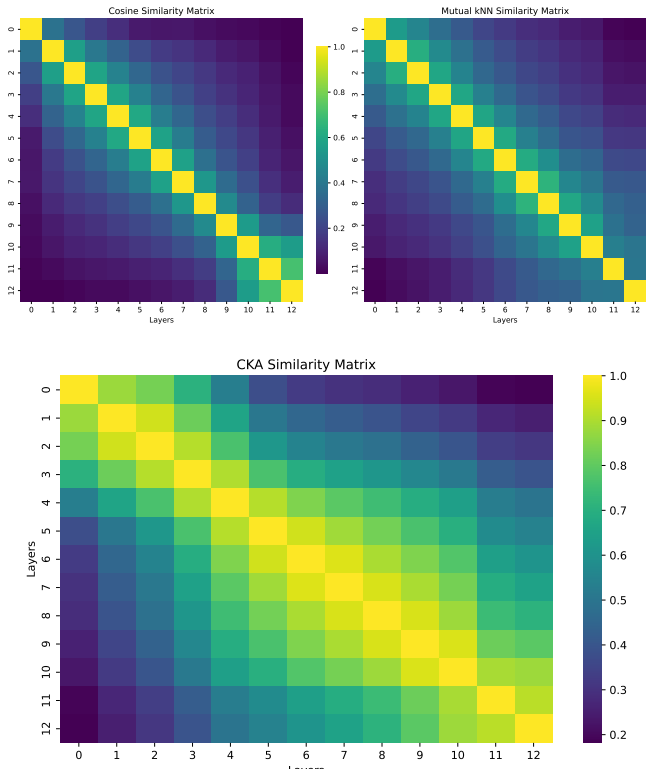


Fig. 4. *Similarity matrix computed using different similarity scores. The layer 0 represents the input of the first transformer layer.*

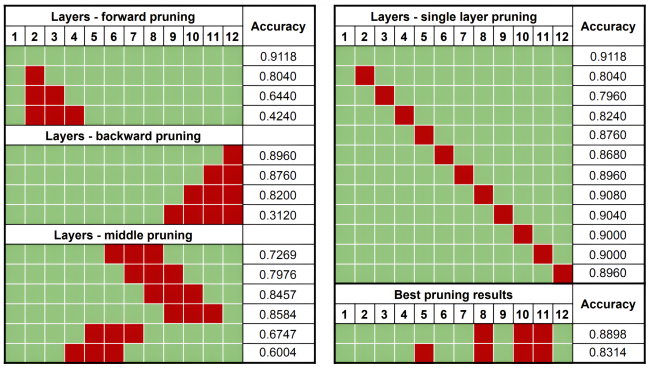
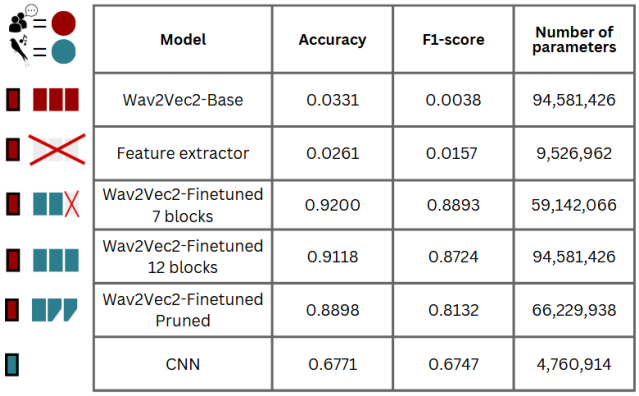


Fig. 5. *Pruning experiment results.*



Model	Accuracy	F1-score	Number of parameters
Wav2Vec2-Base	0.0331	0.0038	94,581,426
Feature extractor	0.0261	0.0157	9,526,962
Wav2Vec2-Finetuned 7 blocks	0.9200	0.8893	59,142,066
Wav2Vec2-Finetuned 12 blocks	0.9118	0.8724	94,581,426
Wav2Vec2-Finetuned Pruned	0.8898	0.8132	66,229,938
CNN	0.6771	0.6747	4,760,914

Fig. 6. *Summary of model performance ans size.*

4.3. Convolutional neural network

Outside of the pruning experiment, we are also interested in understanding how the complex transformer architecture compares with a much simpler one which does not rely on attention. For this reason, we also trained a CNN having two convolutional layer and two linear layers. Due to time constraints we could not perform a fine hyperparameters tuning, therefore it could be possible to achieve better performance with the this architecture.

5. DISCUSSION

In *Figure 6* we compare the results of our experiments. Both the full base wav2vec2 model and the isolated feature extractor are capable of scoring better than the 2% baseline (expected by random guessing the classes). We hypothesise that this is due to the fact that, despite being pretrained on human speech, the model is capable to generalise in small part to other source of sound, as birds song. This is supported by evidence of similarities between the two languages, as described by Berwick et al. [13]. The results of the pre-training pruning experiments show that it is possible to reduce significantly the model complexity (up to 38% in terms of model parameters) while achieving even better performance than the full finetuned model. Similar consideration are also valid for the post-training pruning, even though this approach makes it more difficult to maintain the original accuracy. The first approach definitely provides a better way to reduce a network dimension without sacrificing performance, but has the significant drawback of requiring to finetune a new model every time we want to test a different architecture, which takes large amounts of times and computational resources. The second approach represent a is crucially more scalable workflow as the finetuning process is performed only once (or if the fine-tuned model is provided does not require any), after which the different pruning alternatives only need to be evaluated once on the test dataset. The main weakness of the latter is the need to explore the space of combination, which can become tedious if a trial and error method is used, but this can be overcome by utilising smarter algorithms such as backward layer selection.

6. REFERENCES

[1] Dan Stowell, "Computational bioacoustics with deep learning: a review and roadmap," *PeerJ*, vol. 10, pp. e13152, Mar. 2022.

[2] Jeppe H. Rasmussen, Dan Stowell, and Elodie F. Briefer, "Sound evidence for biodiversity monitoring," *Science*, vol. 385, no. 6705, pp. 138–140, 2024.

[3] Lauryn Benedict and Alan H. Krakauer, "Kiwis to pewees: the value of studying bird calls," *Ibis*, vol. 155, no. 2, pp. 225–228, 2013.

[4] Teresa Dorszewski, Lenka Tetkova, and Lars Kai Hansen, "Convexity based pruning of speech representation models," in *2024 IEEE 34th International Workshop on Machine Learning for Signal Processing (MLSP)*. Sept. 2024, p. 1–6, IEEE.

[5] Dmytro Iakubovskyi, "Bird sounds classification," 2022.

[6] Soumendra Prasad Mohanty, "Sound of 114 species of birds till 2022," 2022.

[7] Xeno canto Foundation, "What is xeno-canto?," 2022.

[8] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," 2020.

[9] Teresa Dorszewski, Albert Kjøller Jacobsen, Lenka Tětková, and Lars Kai Hansen, "How redundant is the transformer stack in speech representation models?," 2024.

[10] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton, "Similarity of neural network representations revisited," *CoRR*, vol. abs/1905.00414, 2019.

[11] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola, "The platonic representation hypothesis," 2024.

[12] Lenka Tětková, Thea Brüsch, Teresa Karen Scheidt, Fabian Martin Mager, Rasmus Ørtoft Aagaard, Jonathan Foldager, Tommy Sonne Alstrøm, and Lars Kai Hansen, "On convex decision regions in deep network representations," 2023.

[13] Robert Berwick, Gabriel Beckers, Kazuo Okanoya, and Johan Bolhuis, "A bird's eye view of human language evolution," *Frontiers in Evolutionary Neuroscience*, vol. 4, 2012.

A. POSTER AND CODE AVAILABILITY

The code for this project is available on GitHub: https://github.com/AndreisMarco/02456_G128_bird_classification