Bird Species Classification: One Step at a Time

Sidhart Krishnan[1], Priya Khandelwal[1] and Rhythm Garg[1]

[1]*Stanford University, 450 Serra Mall, Stanford, CA 94305*

Abstract

Long Short-Term Memory (LSTM) networks can learn long-range order dependencies across time steps, making them particularly well-suited for a variety of use-cases in Natural Language Processing (NLP), such as text generation. We believe that bird species classification can be framed as a sequence prediction problem of taxonomy, where the different ranks can be sequentially predicted to impose more structure on the output space of the model. In this paper, we explore the effectiveness of this novel framework by training and testing an LSTM network on the Xeno-Canto dataset. We compare our model against Multi-Layer Perceptron (MLP) baselines and existing works for audio-based bird classification. Our model outperformed the baselines, achieving 70% accuracy, which serves as a proof of concept that this architecture is expressive enough to be competitive for bird species classification. We conclude that with additional improvements, this method can achieve even more robust performance.

Keywords

LSTM, sequential prediction, taxonomic training, hierarchical structure, bird species classification

1. Introduction

Long Short-Term Memory (LSTM) networks are particularly well-suited to making predictions based on sequences of data. While vanilla Recurrent Neural Networks (RNNs) suffer from the vanishing gradient problem, LSTMs can learn long-range order dependencies across time steps which is why LSTMs have become increasingly prevalent in language modeling[1].

We are curious to see if LSTMs can also be applied to a different domain: bird sounds! All three of us love spending time in nature and want to build an app that can help us identify different birds. Most existing deployed bird classification tools use a semi-automated approach that requires ecologists to have extensive knowledge in signal processing, rendering them impractical [2]. Our model's objective is to take a bird sound audio file as input and then classify the correct species of the bird using extracted features from the audio file.

We believe that bird species classification can be framed as a sequence prediction problem of taxonomy, making the LSTM architecture a good model candidate as we can design architectures similar to sequential text generation models. In particular, the unique insight we believe we have is that the ranks (parts of a taxonomy) can be sequentially predicted to impose more structure on the output space of the model. We explore the effectiveness of this novel application of LSTMs and compare our model against ML baselines and existing works for audio-based bird classification.

CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy

2. Related Work

Prior to the advent of deep learning, Support Vector Machines (SVMs) and Random Forests were the preferred approach to the bird audio classification task [2]. Now, many state-of-the-art solutions for audio-based bird classification are based on fully-supervised deep convolutional neural networks (CNNs) [3, 2]. These approaches typically treat the task as an image classification one by transforming the raw audio into spectrograms. Many researchers fine-tune pretrained ImageNet-based models on spectrograms of bird calls, though some models use more complex architectures like ResNet [4] or BirdNET [5].

Many scholars have also used hybrid RNN-based approaches for the audio classification task. Liu et al. [6] added a Bi-directional Long Short-Term Memory (Bi-LSTM) neural network to DenseNet to extract the temporal correlation features of bird songs, allowing for superior bird song recognition with an average accuracy between 90% and 93%. There are also approaches that weakly label examples which allows them to be robust against background sounds in the audio [7].

However, there did not seem to exist any works that leveraged the sequential structure or hierarchical relationships of taxonomy in bird audio classification, which we believe to be valuable information.

3. Methods

3.1. Dataset and Features

The Xeno-Canto dataset[1] is a collection of 712,477 bird audio recordings over 10,314 different bird species. It is a widely used dataset for the bird audio species classification task and it was used as the primary dataset to train the state-of-the-art BirdNET model[5].

We used 14850 samples across 152 species for our training and validation. These examples were provided through the BirdCLEF 2022[8, 9]. The 14850 samples were split into a $60\% - 20\% - 20\%$ train-test-validation set. Rather than using raw data as an input to the model, we denoise the audio and model the human hearing property at the feature extraction stage. Since humans perceive changes in low frequency sounds with more nuance than they do changes in high frequency sounds, we decided to use the Mel scale to map the audio data to the frequency that mimics what humans would perceive. Using the LibROSA package [10], we extracted the Mel-frequency cepstral coefficients, the spectral centroid, short time Fourier transform, and Mel spectrogram for each audio file. By looking at the label histogram of the dataset shown below, it is clear that there is some heavy class imbalance across the species in this dataset. This motivates us to oversample the train set and val set so that we can train a model that has high precision on the minority species. This results in a completely flat species distribution for the train and val sets.

Additionally, for each model we trained, we worked to prevent overfitting using a validation set and saving the models that perform best on the validation set. We also used this validation set as the performance metric to tune each of the hyperparameters including learning rate,

[1] https://xeno-canto.org/

hidden sizes and dropout probability. Additionally, we had a patience mechanism where if the loss on the validation set increased for 50 consecutive epochs, then training would terminate.

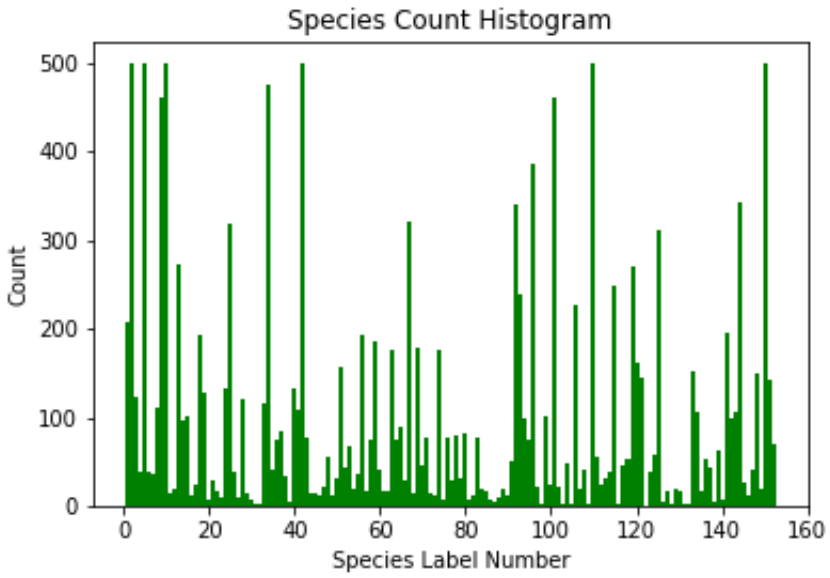


Figure 1: Histogram of Species Labels in Dataset

3.2. Baselines

3.2.1. Multi-Layer Perceptron

The first baseline is a Multi-Layer Perceptron (MLP) model which we used to see if there is a relatively simple non-linear class boundary using the extracted features.

We trained a three-layer perceptron with hidden sizes $512, 128$ using a ReLU non-linearity. This model was trained with Cross Entropy Loss, the Adam optimizer, and a learning rate of 0.001 for 1000 epochs.

3.2.2. Logit-Shifted Multi-Layer Perceptron

The second baseline consists of four Multi-Layer Perceptron models like in the first baseline – one for each taxonomic rank. Then starting from the highest rank – order – we use the corresponding MLP's prediction to boost the logits of the relevant labels outputted by the MLP for the next taxonomic rank, as shown in Figure 2. For example, the logits for families belonging to the order predicted by the order MLP would be increased by a large constant C larger than any other logit in the vector. Then the family prediction would boost the logits for the genuses that belong to that family and so on.

We trained the four MLPs according to the previous section using the same loss, optimizer, hyperparameters and training epochs. However, the output layer sizes for the order, family,

genus and species MLPs were 17, 41, 103 and 152 respectively. For inference, we loaded the best checkpoint for each rank's MLP and began the logit shifting process described above, starting by predicting order and ultimately predicting species.

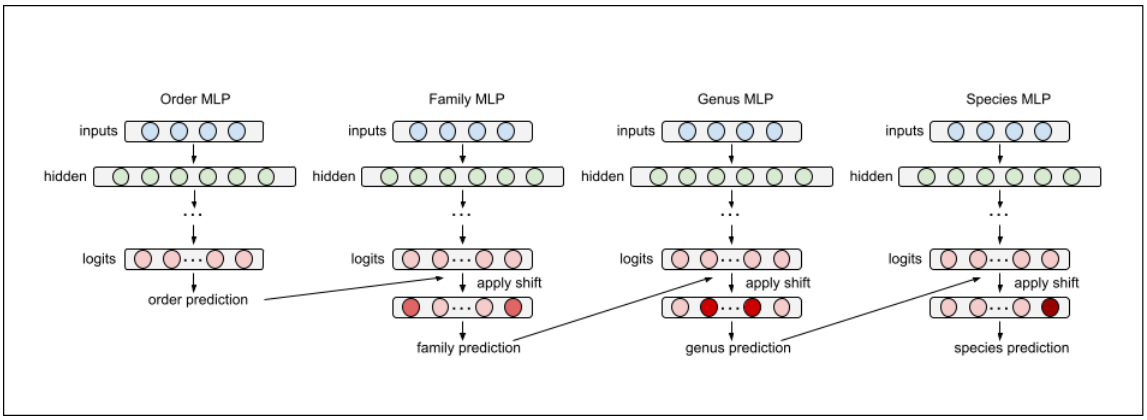


Figure 2: Architecture of MLP Logit Shifting Baseline

3.3. LSTM with Logit-Shifting

Our main model for this project is an LSTM model. A single LSTM cell is shown in Figure 3 where the purpose of the cell is to extract key information about the each input while also saving key parameters over the entirety of the model. The model outputs this information in the hidden state h and the cell state c which are then passed to the next LSTM cell in the sequence.

In particular, if we look at the model architecture in Figure 3, we see that x_0 is the extracted features which is inputted into the first LSTM cell to product a hidden state and cell state h_0, c_0. Then the hidden state is plugged into the linear layers with a dropout layer in between to get the predicted order (i.e. predict the highest taxonomic rank). The soft distribution over the order is then passed in as x_1 to the next LSTM cell which gives a new hidden state and cell state h_1, c_1. Then we pass h_1 into a seperate linear layer, dropout, linear layer architecture to predict logits over the family (i.e. the next taxonomic rank) and so on. We also apply logit shifting mechanism, described in four MLP baseline, to possible logits at each taxonomy and thus reduce the size of the prediction task.

We trained our LSTM using Cross Entropy Loss and the Adam optimizer with a learning rate of 0.001. The loss was calculated for each of the taxonomic ranks since given the true species label, we are able to backtrace to find the true genus, family and order labels to give us losses J_0, J_1, J_2, J_3. Then the overall loss of the model is

$$J = \sum_{i=0}^{3} J_i \tag{1}$$

Notably, during training of the model, we used teacher forcing [11]–a standard technique in text generation models– to mitigate the propagation of training errors forward. We implement

this technique to minimize noisy gradients and better locate sources of error. We achieve teacher forcing by passing in the true probability distribution over the order, family and genus as the x_1, x_2 and x_3 inputs in Figure 3 regardless of the predicted distribution of the model. We also use the true distribution instead of the predicted distribution for logit shifting during training. In this way, each subsequent layer operates under the assumption that the previous layer was exactly correct in its prediction. However, the hidden state of the LSTM still gets passed through to the next layer without any direct tampering.

In terms of experimentation, we tried different model structures using the hidden state of each LSTM cell when predicting each rank. Initially, we started by just having no extra fully-connected layers between the hidden state and the logits used for predicting the rank. We felt like the model was not expressive enough as it seemed to be underfitting (the validation loss was not decreasing), so we then decided to add an extra linear layer followed by dropout layer for robustness between the hidden state of each LSTM and output layer for that rank. The final model parameters we settled on was a hidden size of 512 for the LSTM which is fed into a 512×256 linear layer which undergoes dropout with $p = 0.4$ and then is fed into a final linear layer of size 256 by the output size.

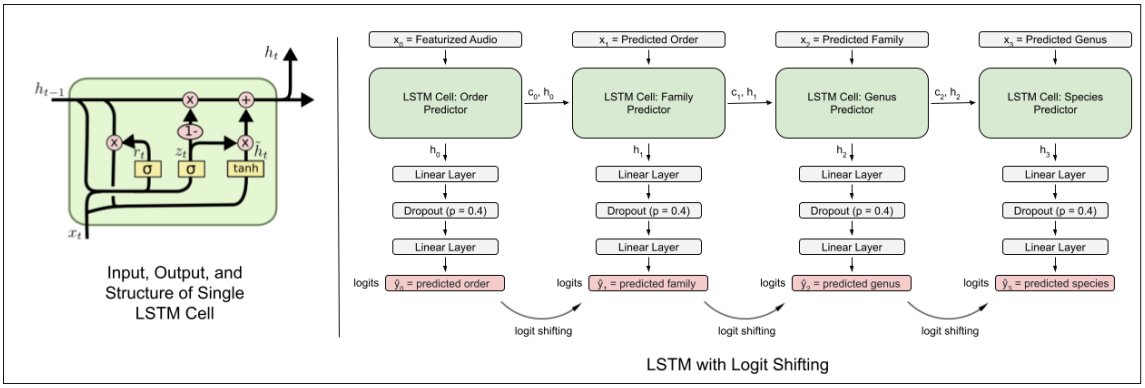


Figure 3: LSTM with Logit Shifting Architecture (including structure of single LSTM cell) [12]

4. Results

Since our task is an imbalanced classification problem, we choose to evaluate our model on metrics besides just accuracy: precision, recall, and F1 score. The performance of each of the models over all of the metrics is shown in Table 1.

4.1. Baselines

For the baseline Multi-Layer Perceptron, the model achieves a test accuracy of 0.57. The accuracy and loss curves begin to plateau around 1000 epochs and we found no significant improvement when trained for longer.

Then, when we compare the Logit-Shifted MLP to the simple MLP, we observed a 4.8

percentage point increase in accuracy, but a decrease in other metrics – namely precision, recall, F1.

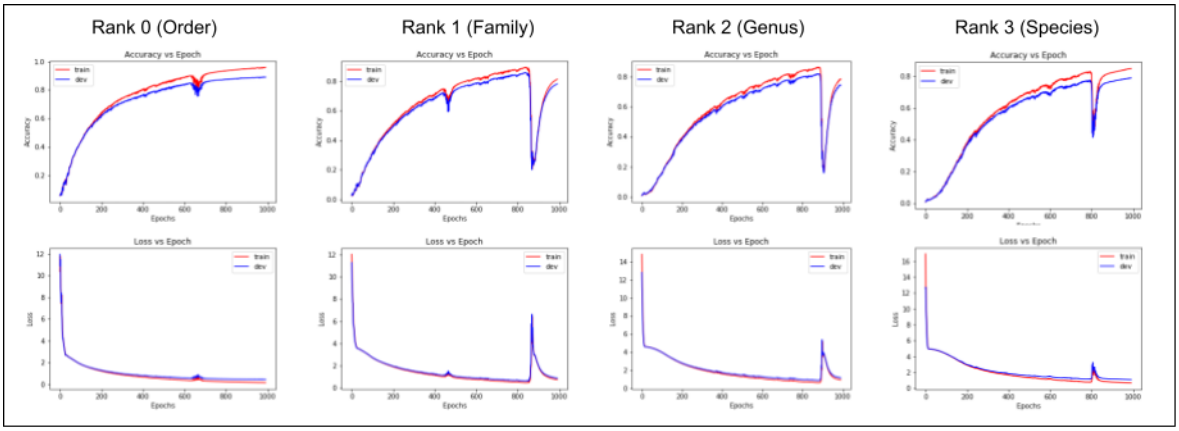


Figure 4: MLP Baseline Accuracy and Loss for prediction at each taxonomic rank

4.2. LSTM with Logit-Shifting

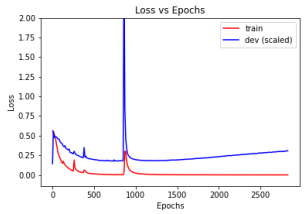
As one can see, as the LSTM progresses to further ranks, the gap between train accuracy and validation tends to grow. This intuitively makes sense since the further along we go in the ranks, the more options there are. Here we also observe around epoch 1000 the negative downstream effects of incorrect order prediction, as can be seen in the train and validation loss downward spikes. We compare the MLP, logit-shifted four MLPs, and LSTM over the key metrics: Accuracy, Precision, Recall and F1 in Table 1 which shows that the LSTM model outperforms both baselines.

Table 1
Model Metrics. The LSTM performs best and points to the sequential prediction paradigm as a viable framework for bird classification.

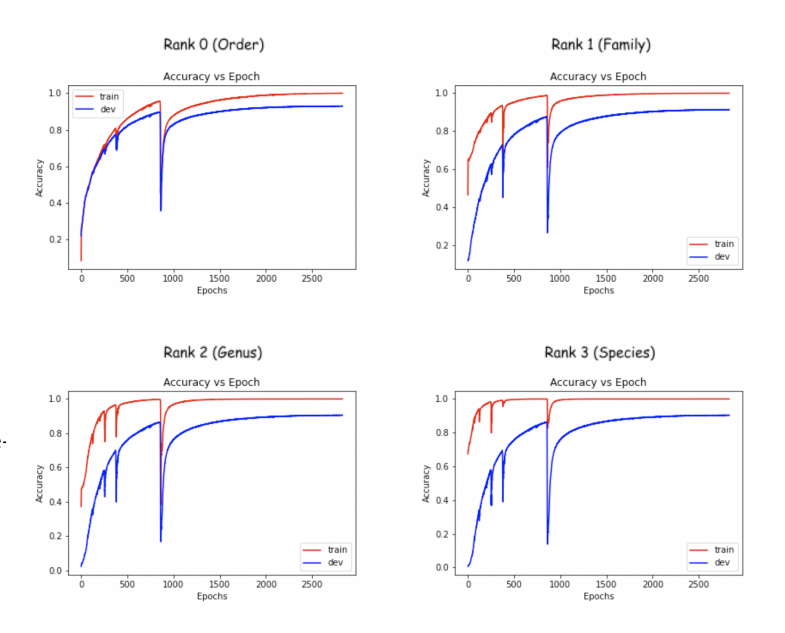
Model	Accuracy	Precision	Recall	F1
MLP	0.5747	0.6286	0.8238	0.6880
Logit-Shifted Four MLPs	0.6229	0.6263	0.6421	0.6096
LSTM with Logit-Shifting	**0.7040**	**0.7006**	**0.8828**	**0.7553**

5. Conclusion and Future Work

In this study, the LSTM architecture outperformed all of the other baselines, likely due to the expressive power of the hidden cells combined with the hierarchical setup for the task. We conclude that this model is a potentially viable approach for the taxonomic sequence prediction task which should be explored further. More broadly, machine learning tasks like



(a) Model Loss for Species Prediction



(b) Accuracy of Model predicting at each taxon level

Figure 5: LSTM with Logit-Shifting Training Curves

taxonomy classification, in which there is some hierarchical structure in the output space, can be interpreted under the same sequential prediction paradigm to reduce the scope of the prediction problem at each level of the hierarchy. Looking forward, we think that the model performance can be improved by adding the LSTM on top of other models; for example, logits extracted from BirdNET could be used as the features inputted to the LSTM.

Another interesting result is with regard to the four MLPs paired with logit-shifting baseline as we saw that the accuracy increased compared to the standard MLP whereas the precision, recall and F1 decreased. This indicates that the four-MLP model is performing quite accurately on majority classes while performing much worse on minority bird species compared to the standard MLP. This could be because biases that occur in predictions at the higher levels of taxonomy get compounded over the course of the prediction. Thus the model's accuracy on minority bird species – which likely fall under minority bird genuses, families and orders – is worse compared to the standard MLP's performance. One future experiment could be to train four different bird classifiers like BirdNET at each taxonomy level and then concatenating them using the same logit shifting mechanism to determine whether these results are unique to the MLP architecture or occur over any bird classification model.

Acknowledgments

Thanks to Stanford University for providing resources and mentorship through the CS 229 class.

References

[1] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780. URL: https://doi.org/10.1162/neco.1997.9.8.1735. doi:10.1162/neco.1997.9.8.1735.

[2] D. B. Efremova, M. Sankupellay, D. A. Konovalov, Data-efficient classification of birdcall through convolutional neural networks transfer learning, 2019. URL: https://arxiv.org/abs/1909.07526.

[3] F. Zhang, L. Zhang, H. Chen, J. Xie, Bird species identification using spectrogram based on multi-channel fusion of dcnns, Entropy 23 (2021). URL: https://www.mdpi.com/1099-4300/23/11/1507. doi:10.3390/e23111507.

[4] M. Sankupellay, D. Konovalov, Bird call recognition using deep convolutional neural network, resnet-50, 2018. doi:10.13140/RG.2.2.31865.31847.

[5] S. Kahl, C. M. Wood, M. Eibl, H. Klinck, Birdnet: A deep learning solution for avian diversity monitoring, Ecological Informatics 61 (2021) 101236. URL: https://www.sciencedirect.com/science/article/pii/S1574954121000273. doi:https://doi.org/10.1016/j.ecoinf.2021.101236.

[6] H. Liu, C. Liu, T. Zhao, Y. Liu, Bird song classification based on improved bi-lstm-densenet network, in: 2021 4th International Conference on Robotics, Control and Automation Engineering (RCAE), 2021, pp. 152–155. doi:10.1109/RCAE53607.2021.9638962.

[7] M. V. Conde, K. Shubham, P. Agnihotri, N. D. Movva, S. Bessenyei, Weakly-supervised classification and detection of bird sounds in the wild. a birdclef 2021 solution, 2021. URL: https://arxiv.org/abs/2107.04878. doi:10.48550/ARXIV.2107.04878.

[8] A. Joly, H. Goëau, S. Kahl, L. Picek, T. Lorieul, E. Cole, B. Deneu, M. Servajean, A. Durso, I. Bolon, H. Glotin, R. Planqué, W.-P. Vellinga, A. Navine, H. Klinck, T. Denton, I. Eggel, P. Bonnet, M. Šulc, H. Müller, Overview of lifeclef 2022: an evaluation of machine-learning based species identification and species distribution prediction, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2022.

[9] S. Kahl, A. Navine, T. Denton, H. Klinck, P. Hart, H. Glotin, H. Goëau, W.-P. Vellinga, R. Planqué, A. Joly, Overview of birdclef 2022: Endangered bird species recognition in soundscape recordings, Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum (2022).

[10] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, O. Nieto, librosa: Audio and music signal analysis in python, in: Proceedings of the 14th python in science conference, volume 8, 2015.

[11] R. J. Williams, D. Zipser, A learning algorithm for continually running fully recurrent neural networks, Neural Computation 1 (1989) 270–280. doi:10.1162/neco.1989.1.2.270.

[12] C. Olah, Understanding lstm networks, 2015. URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.