Convolutional Long Short-Term Memory Neural Networks for Hierarchical Species Prediction

Nithish B Moudhgalya[1], Sharan Sundar S[1], Siddharth Divi[1], P Mirunalini[1], Chandrabose Aravindan[1], S. M. Jaisakthi[2]

Department of Computer Science and Engineering, SSN College of Engineering, Kalavakkam, Chennai, India[1]
School of Computer Science and Engineering, VIT University, Vellore, India[2]
{nithish15066,sharansundar15096,siddarth15130}@cse.ssn.edu.in
{miruna,aravindanc}@ssn.edu.in,jaisakthi.murugaiyan@vit.ac.in

Abstract. Building accurate knowledge of the identity, the geographic distribution and the evolution of organisms is essential for biodiversity conservation. Automatic prediction of list of species is useful for many scenarios in biodiversity informatics. In this work, we propose a hybrid model to predict the species that are most probable to be observed at a given location, using environmental features and taxonomy of the organism. These environmental features are represented as k-dimensional image patches, where each dimension represents the value of an environmental variable, in the neighborhood of the occurrence of the species. The hybrid model Convolutional Long Short-Term Memory Neural Networks henceforth called as CLNN, is a combination of Convolutional Neural Networks(CNNs) and Long Short-Term Memory Networks(LSTMs), where the CNN forms the spatial feature generator while the LSTM focuses on finding the taxonomy. Using the dataset provided by Geo LifeCLEF 2018, the proposed method helped achieve a Mean Reciprocal Rank (MRR) score of 0.003 during the test phase.

Keywords: Niche modeling · Hierarchical embedding · Taxonomic prediction · CLNN

1 Introduction

Environmental niche models have been used by biologists and environmentalists to understand the species distribution in geographic space. These models help reduce resources expended in data collection and analysis, thus giving space for research in analyzing impacts of global phenomenon like climate change, habitat loss, species invasion and evolutionary trends that could help in translocation of species.

Considering the overwhelming uses of species prediction modeling, CLEF organizers posed the Geo LifeCLEF 2018 challenge [5]. The aim of the challenge is to develop a location-based species recommendation system using image-based

representation of environmental features of the immediate surroundings. The main focus was to substitute environmental feature vectors at a given location with image-based representation of the features containing details of the neighborhood. The inclusion of features of the neighborhood better portrays the distribution of species in a region as compared to other niche modeling techniques.

Modeling image-based environmental features involves complex convolutions over multiple filters/channels. Moreover, the impact caused by each feature in determining the likelihood of a particular species can't be analyzed by visualizing the spatial rasters provided. The high number of target classes may lead to vanishing probabilities, an issue where the model's predicted probabilities are low and uniformly distributed across the target classes, thus making the learning process error-prone.

To overcome these challenges, we propose the use of Convolutional Long Short-Term Memory Neural Networks (CLNN) architecture shown in Fig. 4, to model the species distribution given their spatial environmental features along with the species taxonomy. The introduction of taxonomy addresses the vanishing probabilities by reducing the target classes.

2 Methodology

The proposed CLNN model is a hybrid pipeline of CNN and LSTM layers. Both CNN's and LSTM's are brainchildren of Deep Learning Techniques. Deep Learning (DL) is a broader type of machine learning algorithms, drawing inspirations from the biological nervous system. In DL, a cascade of multiple layers of non-linear processing is used for feature extraction and transformation. Convolutional Neural Networks(CNNs) [2] are a class of deep neural networks that are used for Computer Vision or analyzing visual imagery. CNN makes use of a set of learnable filters, which are used to detect the presence of specific features or patterns present in the original image. Different filters which detect different features are convolved and a set of activation maps are produced. These maps are then flattened i.e. reduced to a n-dimensional vector, and fed into the LSTM. Long Short-Term Memory Networks(LSTMs) [4] are a class of deep learning techniques that are constructed based on recurrent concepts in neural networks. The LSTM cells have 3 gates namely input, output and forget that helps it to arbitrarily remember some input thus giving it memory. They are usually used for modeling sequences and to predict the change in patterns with respect to some fixed variable. The LSTM layers used here, get the flattened features from CNN and find the taxonomy of the species as a sequence. The taxonomy of organisms was used to capture the intra rank similarities and also reduce the search space for species prediction. Embeddings were added to represent the labels in k-dimensional space and to maintain their taxonomic context.

2.1 Use of taxonomic nomenclature

Taxonomy of organisms was introduced in Biology, to group species based on their common ancestral characteristics. According to Darwins Common Descent theory [1], the process of speciation occurs due to the adaptation of organisms/species to environmental changes. This results in different species with a common ancestor, that share common characteristics and requirements. This fact has been exploited in our species prediction model by making use of taxonomic hierarchy. The Fig. 1 shows the radial tree of taxonomic hierarchy, where the center of the diagram is the root node "NULL" and the leaf nodes represent the species_ids. The diagram shows that many labels get eliminated as we traverse radially outward, thus narrowing our search space greatly in reaching the right species id. In Fig. 2, the bar plot shows that the number of class labels at the final layer during classification, dropped from 3336 to about 72 with the introduction of a hierarchy of 5 levels. The taxonomic ranks were contained in the below mentioned data columns - Kingdom, Phylum, Class, Order, Family, Genus and Species_glc_id. As the first 2 ranks were same for all the given instances, only the last 5 levels were used in the tree structure shown in Fig. 1.

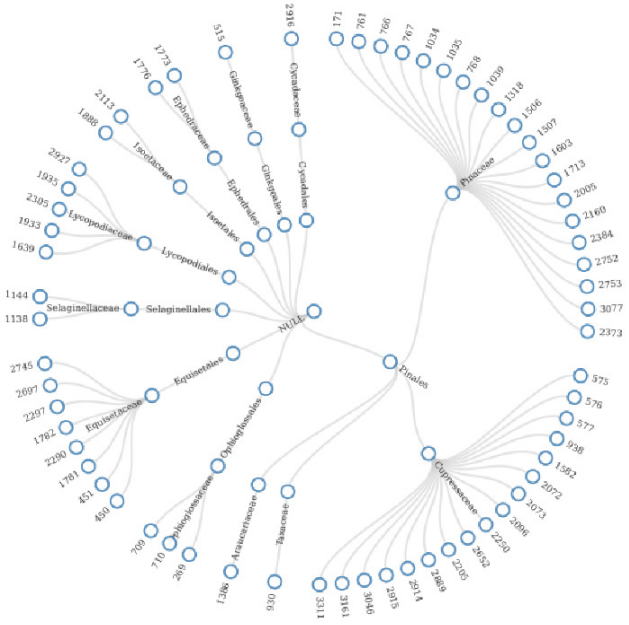

Fig. 1. Taxonomic Hierarchic Tree

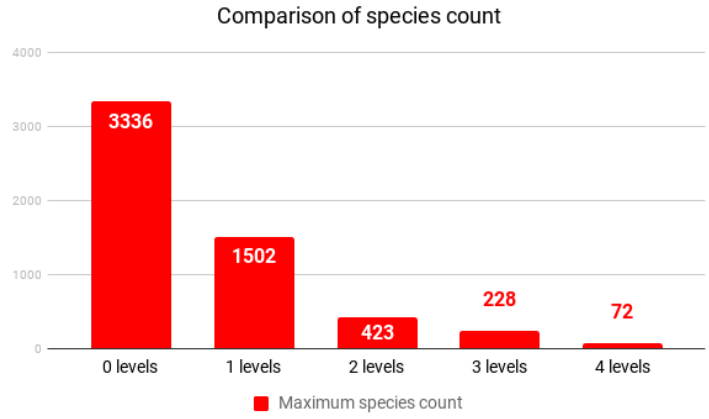

Fig. 2. Bar plot showing reduction in number of species labels with taxonomic hierarchy introduction

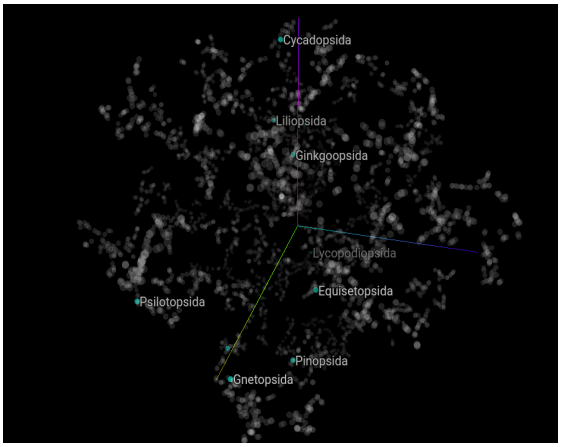

Fig. 3. t-SNE plot showing the Póincare Embeddings of the taxonomic labels of the data

2.2 Embedding

Embeddings were introduced in Machine learning especially in Natural Language Processing (NLP) WordtoVec algorithms, by Tomas Mikolov [6]. Many embedding algorithms have since been developed, but most of them use euclidean distance as the measure of similarity. However, since hierarchical vectors

form a tree structure which resembles a hyperbolic curve as seen in Fig. 1, using hyperbolic distance as the measure of similarity embeds species vectors aptly. Póincare embeddings [7] can be used represent hierarchical vectors. The Fig. 3 shows a t-SNE plot [9] of the Póincare embeddings created in experiments. The named dots are the class names and the other indistinct dull spots belong to each unique label in other lower level taxonomic ranks. The axes shown in the plot are used to visualize the n-dimensional vectors in a 3D space but are not correlated to any coordinate system. From the plot, it can be inferred that the class labels are embedded far apart and the corresponding lower ranks are clustered along, thus preserving hierarchy.

2.3 CLNN architecture

The CNN used is the state-of-art ResNext model architecture [10] that combines Inception [8](parallelized convolutions) and Resnet [3](sequential layers with residues) together. The first + symbol in the Fig. 4 shows the global average pooling of 256 parallel convolutions, while the second + represents the concatenation of input residue to the output from that block. CNNs are used to extract meaningful spatial features from the given tiff images. These features are repeated over 5 time steps and passed on to the LSTM layer to predict the taxonomic ranks. At each time step, the LSTM predicts the taxonomic ranks namely class, order, family, genus and species as shown in Fig 5. The 3 Dense layers containing [128,128,5] neurons respectively, follow the LSTM and are time distributed, which ensures that the logcosh loss is calculated and the errors in Póincare embedding predictions are back-propagated to both the LSTM and CNN layers with equal weights for each time step. This helps the CNN work with the LSTM, and provide features that help the LSTM improve its predictions. No pre-trained weights were used and hence the model trains entirely on the data provided.

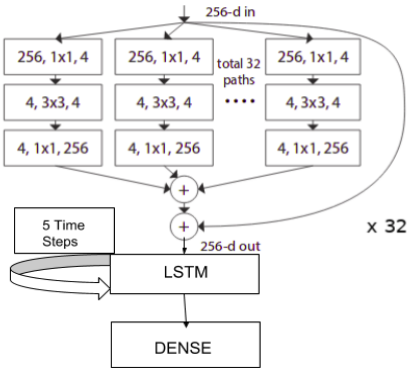

Fig. 4. CLNN architecture

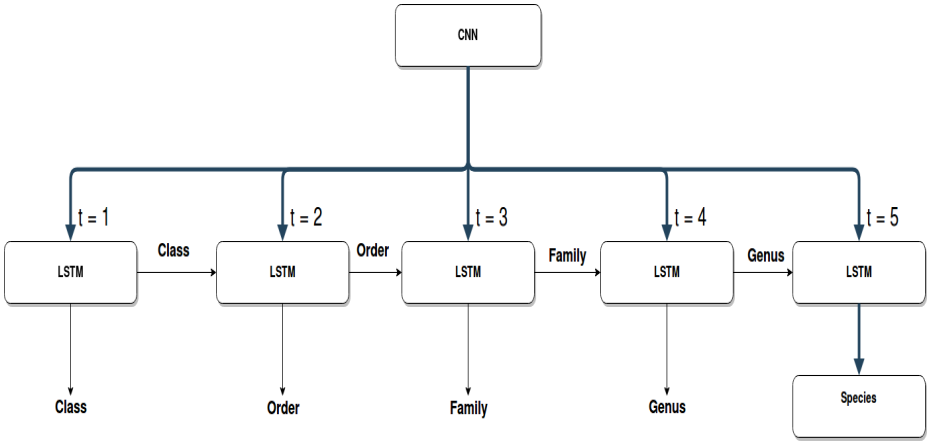

Fig. 5. LSTM time-step predictions

3 Experiments and Results

The GeoLifeCLEF 2018 dataset, consists of 2,18,000 tiff images, with each tiff image containing 33 raster layers of environmental features. The batch size for all experimental purposes was fixed at 32. The CLNN model proposed contains 23M trainable parameters and 32,000 non-trainable parameters. The learning rate of the model was set to the default value of 0.001. The codes were written in Keras API with back-end as tensorflow. Python packages like numpy, scikit-io, tifffile, PIL and pandas were used for data preprocessing. The resource configurations include 4GB dedicated graphics by Nvidia GEFORCE 840M processor and 12GB CPU memory. Different combinations of main concepts mentioned earlier were used to make the following runs.

SSN_1 The CLNN model used the given 33 layers of spatial feature maps to predict the taxonomic ranks of species. Every layer of each tiff image was first center-cropped to a size of 32x32 and then fed to the model which ran for 5 time steps classifying the image into taxonomic ranks. However the model was not trained to classify each rank .i.e. the back-propagation algorithm ran only for species classification and not for the other ranks. Adam optimizer was used to find the minima of the sparse categorical cross-entropy loss function.

SSN_2 The concept of embeddings was introduced and an independent model was used to create the embeddings of 10 dimensions between each pair of taxonomic ranks .i.e. class labels were embedded to find order, order labels embedded to find family and so on. The embedded vectors were used as identifiers of the unique labels in the CLNN model. So the architecture was modified to predict a 5 time-stepped sequence of 10 dimensional vector, where each vector corresponds to its unique taxonomic rank labels. However, these embeddings did not capture the context of hierarchy, which was not used for creating them.

The back-propagation algorithm runs only for the last time step .i.e. the species predictions. Adam optimizer was used to find the minima of a MSE loss function.

SSN_3 The concept of Póincare embeddings was used and binarization was used to convert each ordinal feature layer into n-1 layers, where n stands for the number of categories it can assume. All images were fed into the model with the original dimensions of 64x64. The CLNN model was made to learn each level in the taxonomic hierarchy, by adding a time distributed wrapper around the layers following the LSTM. The model predicts the Póincare embedded vector of 5 dimensions, at each time step for a particular image which is then decoded to find the corresponding labels. The ranks of species were calculated based on the distance between the learned embeddings and the model predicted embeddings of the species. Since Póincare embeddings was used, the logcosh loss function was used with Adam optimizer by the model for a batch size of 32 tiff images.

SSN_4 All images were fed into the model with the original dimensions of 64x64. A time distributed wrapper was added to the layers following the LSTM segment of the CLNN to ensure the back propagation algorithm applied to every time step, thus enabling the model to learn the entire taxonomic hierarchy. Again, the concept behind binarization was applied to ordinal feature layers. The Adam optimizer was used to find minima of sparse categorical cross-entropy loss function, as the final outputs were one-hot encoded.

The metric used to measure model efficiency is Mean Reciprocal Rank(MRR) which is calculated as,

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i}$$

The results are calculated for both training and testing datasets and shown in Table 1.

Table 1. Accuracy and MRR of each run

Run Name	Training set accuracy	Test set MRR
SSN_1	0.15	0.0004
SSN_2	0.22	0.0013
SSN_3	0.38	0.0030
SSN_4	0.25	0.0016

4 Conclusion and Future Work

The hybrid CLNN model with the power of taxonomy resulted in low accuracies initially but showed promising surge in the later stages(0.004 to 0.0030). The use of Póincare embeddings along with learning taxonomy at each time step, showed best results so far. However, these relatively low values can be attributed to some

or all the following reasons. The LSTMs need a large number of epochs to learn the sequences but due to the processing and resource bottlenecks, the LSTM was trained only for a few epochs. The mathematical complexity involved in incorporating Póincare Embeddings for the taxonomic prediction is still debatable. Thus fine tuning the hyper parameters of the CLNN model to maximize the use of taxonomy and embeddings can be incorporated in future. The use of embeddings at output levels are hard to model as they are n-dimensional float values for each label and cannot be easily predicted by model within a few epochs thus displaying huge errors at starting stages. To find the top n ranks of species, the distance between model predictions and learned embeddings were compared. As the embeddings were calculated using different functions and CLNN trained on different loss function, the distances calculated need not belong to either coordinate system thus giving curious results in some cases. Also the shuffle among patch_ids and species_ids predicted by model may be attributed to bottlenecks in CPU and GPU computations owing to the use of Sequence generator.

CLNN can be modularized in the future, by training the CNN and LSTM separately, to avoid misleading gradient problem, wherein the errors made by LSTM need not be reflected into CNNs feature generations. The model would thus function like image-captioning with CNN features being fixed and LSTM training to understand sequences from these fixed features. Yet another family of thoughts could give rise to Branch-CNN [11] in which the coarse layers are used to predict lower level hierarchy and finer layers to predict higher level hierarchy. Each branch trains specifically for the corresponding taxonomic rank thus compartmentalizing the CNN's features generated.

References

1. Darwin, C.: The Origin Of Species. John Murray (1859)
2. Deshpande, A.: A Beginner's Guide To Understanding Convolutional Neural Networks. `https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/` (2016)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR **abs/1512.03385** (2015)
4. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (Nov 1997)
5. Joly, A., Goëau, H., Botella, Christophe, G.H., Bonnet, P., Planqué, R., Vellinga, W.P., Müller, H.: Overview of lifeclef 2018: a large-scale evaluation of species identification and recommendation algorithms in the era of ai. In: Proceedings of CLEF (2018)
6. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26, pp. 3111–3119. Curran Associates, Inc. (2013)
7. Nickel, M., Kiela, D.: Poincaré embeddings for learning hierarchical representations. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 6338–6347. Curran Associates, Inc. (2017)

8. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. CoRR **abs/1409.4842** (2014)
9. Wattenberg, M., Vigas, F., Johnson, I.: How to use t-sne effectively. Distill (2016)
10. Xie, S., Girshick, R., Dollr, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. arXiv preprint arXiv:1611.05431 (2016)
11. Zhu, X., Bain, M.: B-CNN: branch convolutional neural network for hierarchical classification. CoRR **abs/1709.09890** (2017)