Quinnipiac UNIVERSITY

Evaluating the effectiveness of utilizing morphological characteristic input through a neural network to classify fish gut flora

Alexander Kirst, Dr. Jonathan Blake[†] , Dr. Mark Hoffman[†], and Dr. Lisa Kaplan[‡] .
[‡]Department of Biological Sciences, [†]Department of Computer Science, 275 Mount Carmel Ave Hamden, CT

Abstract

Identification of an organism's residential gut flora can be a time consuming and costly proposition involving purification, PCR, sequencing, and BLAST searches. The focus of this project is to reduce both cost and time by developing a neural network to identify microbes based on their morphology. A Python-based neural network library was developed and used to classify *F. heteroclitus* gut flora based on morphological characteristics ascribed to the microbial colony (color, size, appearance, and convexity) as well as those associated with the individual bacterial cells (shape and gram stain). Using previously identified samples with known morphological data, this project aims to train a neural network for use in identifying unknown bacteria samples. The results of this study indicate that the network is accurate at predicting the identity of bacteria within the Vibrionaceae family, and shows promise for others. While additional data is needed to expand predictions to other families, development and application of this neural network should save time and limit the expense associated with PCR, generating 16S ribosomal sequences, and database searches.

Introduction

The costs (in time and money) to identify bacteria isolated during experiments can be high. This project investigates the use of an Artificial Neural Network (ANN) to classify bacteria based on the physical characteristics of the colony, as well as characteristics easily obtained through experimentation. These characteristics are described in Table 1. ANNs are excellent tools for mapping one set of values to another, or for classifying patterns [2]. An ANN is a collection of "neuron-like" nodes connected in an acyclic, weighted digraph, where the nodes are clustered into layers. The nodes in a given layer are fully connected to the nodes in the neighboring layers, and are not connected to each other. The first layer is designated as the input layer, encoding a pattern to be classified, and last layer is designated as the output layer, representing the classification. All other layers are hidden (Figure 1). Each node has an activation (stored as a single real value) that is determined from the sum of all its inputs. The weights for the edges in the ANN are initially random. A series of input/target training patterns are presented as inputs to the ANN, and the edge weights are modified (using gradient descent). After sufficient training, the weights of the edges of an ANN encode the classification of inputs to outputs.

Materials & Methods

Input/target patterns were determined from prior sequencing experiments investigating fish microflora. 345 bacterial samples were obtained, each consisting of the full set of morphological identifiers. These identifiers were converted to binary strings to represent the input patterns to the Artificial Neural Network (ANN). Each input pattern has a corresponding target output. These outputs are binary strings encoding particular species of bacteria.

An ANN was created in Python using the Keras deep learning library [4]. The ANN has 55 input nodes, 64 hidden nodes, and 78 output nodes. All training and testing was carried out on a 8-core laptop computer.

Training (a one time expense) with 10,000 epochs and a batch size (number of training samples in one epoch) of 25 takes less than one minute, and predictions are almost instantaneous. Figure 3 shows the total error for the ANN results as the training is carried out.

Results

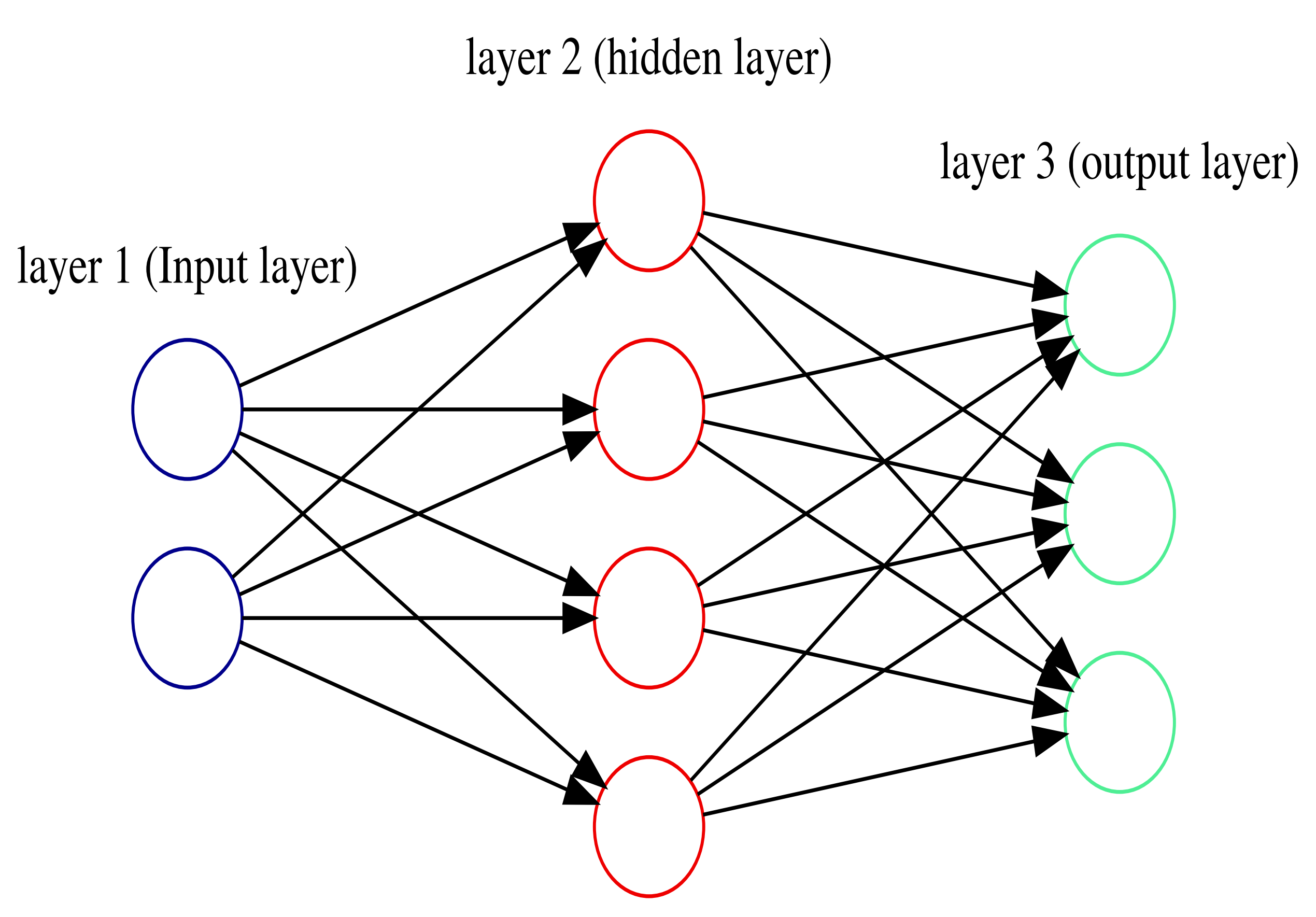

Figure 1. An ANN (Artificial Neural Network) This diagram shows a simple example 3-layer ANN. For this study, the input layer encodes the morphological characteristics of the bacteria, and the output layer represents predicted species.

Color	Shape	Size	Convexity	Appearance	Gram Stain
Yellow	Bacilli	0.5mm	Convex	Dull	+
Orange	Cocci	1mm	Raised	Opaque	–
Light Orange	Coccobacilli	2mm	Flat	Shiny	
Cream	Diplococci	3mm		Transluscent	
Beige	Rods	4mm			
Brown	Skiny Rods	5mm			
Clear	Tiny Cocci	Punctiform			
Tan	Tiny Coccobacilli	Super Punctiform			
White	Tiny Diplococci				
	Tiny Rods				

Table 1. Morphological Characteristics This table contains all morphological data used as input nodes to the ANN.

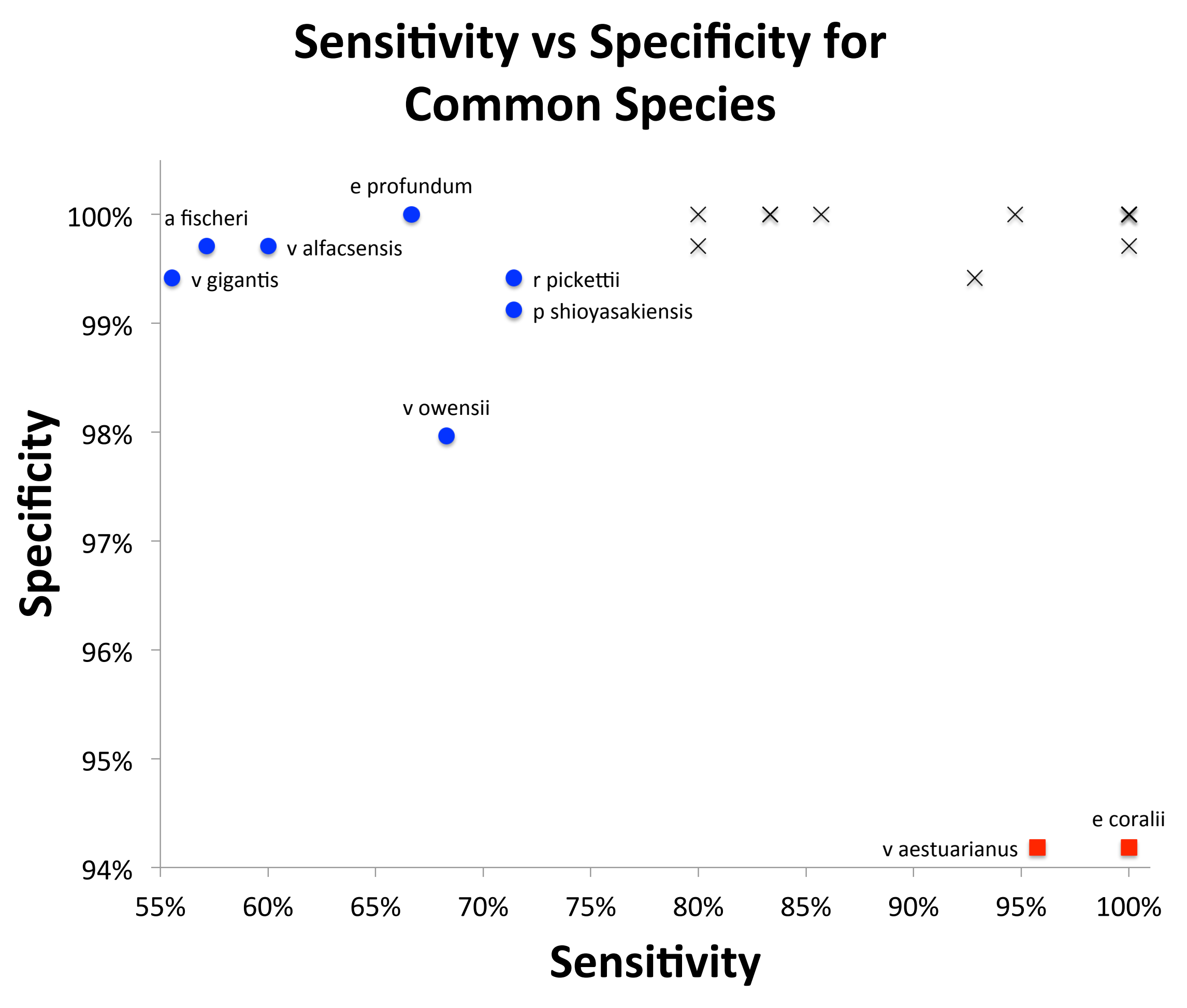

Figure 2. Sensitivity vs Specificity Chart This chart shows the **sensitivity** (True positive rate: Percentage of true positive matches correctly identified) and **specificity** (True negative rate: Percentage of mismatches incorrectly identified) for the species with the highest number of occurrences in the training set.

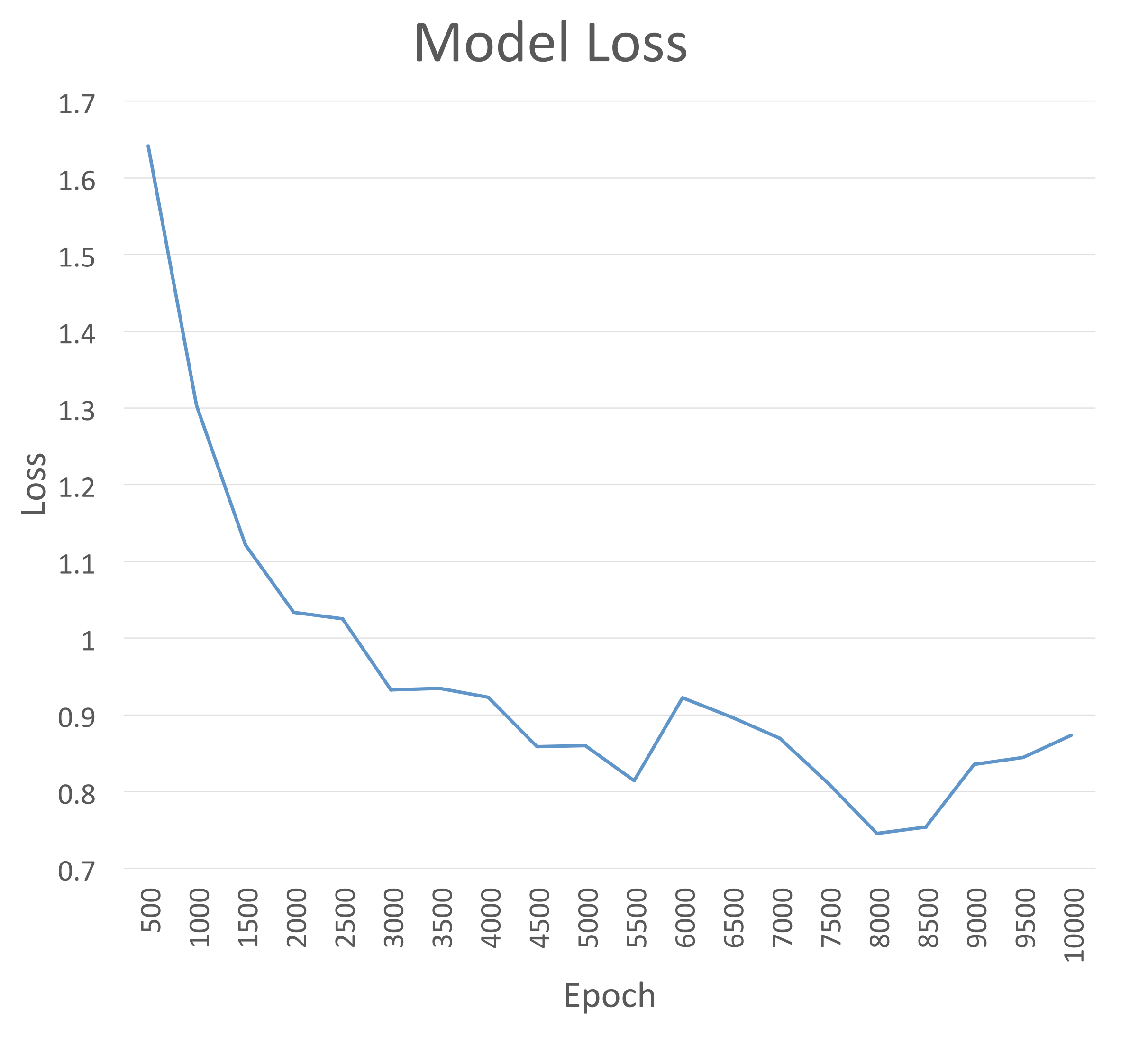

Figure 3. ANN Loss Loss is a scalar minimized during our training of the model. The lower the loss, the closer our predictions are to the expected output.

Discussion

The results show that the network implemented was able to correctly predict 80% (277/345) of the species in our data set. When considering just genus or family, this number increases to 87.5% and 94.2% respectively. ANNs can have difficulty classifying inputs that are underrepresented in the training set due to a lack of training patterns seen. This is particularly evident in our data set when considering species represented fewer than 5 times. Those species have an average sensitivity of less than 70%. Several observations can be made about the results for those species occurring 5 or more times: The training set is heavily weighted towards species from the Vibrionaceae family. This explains the lower specificity for the highest occurring species in Figure 2 (as the network over-learns the patterns for Vibrionaceae); Of the remaining species, 7 have a sensitivity less than 75%. These species are candidates for further analysis. One in particular (*v owensii*) has a very low sensitivity (68.3%) which is surprising considering it is the second most common sequence (41 occurrences) in the training set.

Citations

1) Eberhart, Russell C. *Neural Network PC Tools: A Practical Guide*. San Diego: Acad., 1990. Print.
2) Heaton, Jeff. *Introduction to Neural Networks with Java*. St. Louis: Heaton Research, 2005. Print.
3) Python Software Foundation. Python Language Reference, version 2.7. Available at http://www.python.org
4) Chollet, François. Keras Documentation, version 2.02. Available at http://keras.io

Acknowledgements

I thank Hannah Kissinger, Gabriella Finello, Megi Gracari, Erica Kleinbaum, Nadia Jarvis, Matt Stelinle, and Ali Luchini for their contributions of morphological data, as well as the faculty and staff of Quinnipiac University for their ongoing support, especially Dr. Lisa Kaplan, Dr. Jonathan Blake and Dr. Mark Hoffman.