Braden Eichmeier
CS 5600
Project 1
October 20, 2018

Classification of Images and Audio Samples with Artificial and Convolutional Neural Networks

Introduction:

Beekeeping requires routine maintenance from the beekeeper to the beehives. When a beekeeper must conduct maintenance, they must vacate and pacify the bees by exposing them to large amounts of smoke. After maintenance is performed, a period of time must pass before the hive returns to regular activity. This recovery time is important for the beekeeper to obtain another metric to measure the hive's welfare. This project utilizes neural networks to detect the presence of bees from both audio and visual data. This classification may then be used to inform the beekeeper of regular hive activity.

Image Classification:

Training a network to identify bees in a 32 x 32 image required slight modifications from the network setup required for the MNIST dataset. Upon training networks using the functions, cost functions, and optimizing procedures presented in class, results for bee identification returned no better than a random guess. During the testing procedures, the validation accuracy and validation cost remained constant. This occurred as a result of the learning rate being too high and the mini batch size being too low. As the network attempted to learn, each weight adjustment overcorrected from the desired path. By increasing the mini batch size, the randomness in the adjustment steps becomes reduced, which leads to a less random correction in each step. By reducing the learning rate, the adjustment step reduces and avoids overshoot.

After multiple network revisions, the first better-than-guessing network occurred as a result of using the Adam optimizer. Using stochastic gradient descent to optimize the system never led the net to converge on favorable results. Changing to the Adam optimizer increased the validation accuracy to about 70%. Further improvements occurred by experimenting with the fully connected layers: altering the number of nodes in each layer, adjusting the number of hidden layers, and experimenting with different activation functions. The 'elu' function in the hidden layers and a softmax in the output layer produced the best results in the ANN.

Learning from optimizing the ANN resulted in quickly training the convolutional net. Experimenting with the CONV net showed favorable results using relu functions in all of the convolution layers, sigmoids in the fully connected layers, and softmax in the final layer. Table 1 shows the validation results for both nets. The file 'training_nets.py' includes the network architecture and training procedures for both networks. Training accuracy for both nets exceeded 98%

Table 1. Validation results for both networks on classifying bee images

Network	Bee Accuracy	No-Bee Accuracy
Image ANN	88%	90%
Image Conv	94.5%	92.5%

Audio Classification:

Classifying audio samples to detect bees buzzing, crickets chirping, or random background noise proved more difficult than image classification. Designing and training a network to accurately perform this classification seems to depend heavily on the preprocessing techniques employed on the data. I employed two major variations of data preprocessing, and numerous iterations within each variation.

The first attempt at preprocessing the data included taking segments of audio samples within the whole file. Several audio samples were taken from each file in order to artificially increase the amount of training data. This technique resulted in very unstable network training sessions. The networks experienced one of three outcomes: random guessing with 33% accuracy, non-convergence, and a distribution function behavior. When the network seemed to begin learning, the validation accuracy began at 33%, then increased to about 54% accuracy before returning to about 40% accuracy.

The second preprocessing method employed the resample function in the scipy.signal library. This function resamples the entire audio file to a certain length. Employing this function results in steady accuracy improvement to about 48-55%. Despite this, no better results were achieved despite testing approximately fifty different networks. Testing the networks on the individual audio classifications created the results in Table 2. The data shows the networks perform exceptionally well in cricket and noise data but produce abysmal results in bee classification. Because the bee results are much worse than guessing, the network architecture learned to not identify a bee buzzing. In most of the bee files, the network predicted they were cricket files. For both networks, the training accuracy exceeded 90%.

Table 2. Validation results for classifying three types of audio files

Network	Bee Accuracy	Cricket Accuracy	Noise Accuracy
Audio ANN	0.04%	99%	67.6%
Audio Conv	0.1%	95.6%	65.4%

Conclusion:

Machine learning produces accurate results in identifying whether a bee is present in an image. In order to produce positive results, training requires a mini batch size of about forty and a learning rate about $O(10^{-4})$. Misalignment of these hyperparameters results in a validation accuracy equal to a random guess. Applying the same principles to audio classification performs well in classifying crickets and ambient noise, but performs well below guessing in identifying the sound of bees buzzing.

Researching the difference between the two noises show bees buzz at about 500 Hz; crickets chirp at about 4 kHz. This drastic difference in frequency likely explains the networks poor performance. Creating a network to identify all three audio sources would require a preprocessing technique that promotes identification of both high and low frequencies.