ECE/CS/ME 539 Introduction to Artificial Neural Networks

Project Progress Report

Dog Breed Classification

Team #: 19

Team members:

Jack Hanfland, Computer Science Undergraduate

Date of submission: 11/18/2023

ABSTRACT

So far in this project, I have combined elements from existing dog breed classification models and our existing animal classification from earlier to build a superior one. I added transformations to the images like the pre-trained model from Kaggle that had great accuracy and implemented a stop if not improving to the training. I've tried implementations with and without pre-training. However, from my experience, without a pre-trained model, my accuracy takes much longer to become good. The Resnet18 model starts fairly accurate for the classifications and improves about the same amount per epoch until it nears its peak accuracy.

2. Introduction: Overview: This project will use images from a Kaggle dataset of dogs of different breeds to train a model to predict which dog breed is in the image.

Background: Dogs are the most common pets in the world, and there are many different breeds based on their genetics. Dog breeding has been especially popular in the last century, and new hybrid breeds have been created. Knowing a dog's breed is essential, as it correlates to the dog's personality, behavior, and size. Making a model that could help classify these dogs would be important to buyers of dogs who are unsure of what breed they are looking at. Being able to know the breed of the dog could influence their decision to find a better-suited pet. Having a high-accuracy classifier would allow buyers to be sure of what breed their dog is or what breeds it could be mixed with.

3. Method:These classifications are each unique and provided me insight into how I could add features from each to build a better model. I experimented with layers from the pre-trained model in the first link, added transformations to make the images easier to classify, and cut the training if the model is no longer improving. I developed my neural network from scratch but I used elements from the existing code from users on Kaggle and existing code from work I have done in this class. I am using the cross-entropy loss as the optimizer and the loss as the mean square errors for my model.

4. Results: I edited my program today and am now unable to get my results due to sheer time it takes to run the model on my laptop gpu. I am working to find ways to improve runtime but it takes over an hour for each epoch currently. This is similar to the time it takes for running any of the code from the models made from people on Kaggle, so I know it is not just my code. However, before some of the edits I made today, I remember achieving an accuracy of around 85% after 5 epochs running earlier this week. This accuracy is below the accuracy I am aiming for. This means I need to improve on my model to get up to the baseline results of 90%.

5. Discussions: Based on my results I am sure that my model is close to where it needs to be, but it definitely needs improvement. I am working on researching new ways to preprocess the images, to improve the accuracy but also reduce the size. My first step is making the program a good amount faster. I have been struggling to determine if my edits are actually improving anything because of the amount of time to get new results after. This has made the improvement process much slower and more difficult. After I am able to run it more efficiently, I should be able to further tweak and improve my code to achieve an accuracy above my goal of 90%

References: https://www.mathworks.com/help/deeplearning/ref/resnet18.html