Assignment 7 - Image Processing with CNN

Executive Summary:

Our assignment is to evaluate machine learning technologies and tools for automatically labeling images provided by the end user. We need to advise and provide the website provider with guidance on the best image classification techniques that result in the highest accuracy scores for image classification. As part of the study we will explore various factors in the design of CNN (Convolutional Neural Networks) and other neural networks, such as numbers of layers and nodes within layers, types of layers (convolutional or pooling), and other hyperparameters in a systematic manner using factorial experimental designs.

Research and Design:

We utilized data from a Kaggle competition in 2013 - the original data set consisted of 25,000 images of cats and dogs. For our assignment, we are utilizing a small subset of this data comprised of 2,000 images: 1,000 cat images and 1,000 dog images to reduce model training time. The data will be split up into an 80% (train) and 20% (Test) setup for modeling purposes. A number of models were built using different neural network architectures and classification accuracy of the model was recorded. Runtimes were also tracked, but the focus of the assignment was returning the highest accuracy.

Technical Overview:

We used Python's Keras API within the TensorFlow package for implementing Convolutional Neural Networks on the cat and dog dataset. It's important to recognize upfront the dimensions of the image, 64 x 64, we are dealing with as it's needed for building out executable networks that will run without errors. Images are grayscale where channel is set to 1 as last piece of ranking where we have (image #, image size, image size, channel) as final model shape. For testing purposes Y scores and model accuracy can be either one of two values whether or not it correctly predicts the image as a cat or a dog . Models can then be built and compiled according to our specifications using training and test set accuracy score and processing times as the basis for model comparison. We established a fairly high number of epochs in order for the model to learn as much as possible, given the limited number of training data and the quality of it. Our focus was to choose large enough epochs where test accuracy would hit its peak value and not improve despite training accuracy continuing to rise (overfitting of data). We started with the AlexNet and LeNet architectures as our base models. We then experimented by systematically changing one parameter at a time to pursue the highest test accuracy possible. We also explored more advanced deep learning techniques, data augmentation -- adding

additional images through cropping, rotating, zooming, flipping, etc. to see whether it would help improve accuracy scores on the test dataset of the model.

Findings:

To arrive at the best model that resulted in the highest accuracy score, we ran 19 different models in an experimental design setting, changing various hyperparameters and the base architecture one step at a time. Across all the models (refer to the attachment with code), model 1 achieved the highest accuracy score of 0.71 on the test data - this model was built on AlexNet architecture, with convolution, max pooling, convolution, max pooling, convolution, and one dense output layer. It involved a kernel size of 3 and 2 layers. Despite going through an experimental design changing many hyperparameters one at a time, applying data augmentation and even trying a different architecture type we were not able to produce a higher test accuracy. See attached code in zip file for thought process and steps taken to produce highest accuracy model.

Conclusion and Recommendation:

To conclude, we identified the best model to arrive at the highest accuracy rate of 0.71 based on the partial data set that was provided. We noticed that keeping the model simple with 2 layers and kernel size of 3 produced the best layer where adding more nodes and layers only hurt the model accuracy. Suggested enhancements to increase model accuracy further include utilizing the full data set of all images since training on a small partial dataset is prone to overfitting. Also exploring color images instead of grayscale and using higher resolution images should help improve model prediction in accurately predicting whether the image is a cat or dog.