Summary

Convolutional neural networks (CNNs) is a class of deep neural networks and is most popularly used for image recognition and analysis. The design of CNNs - consisting of convolutional layers, pooling layers, and fully connected layers – is highly favored when dealing with higher resolution images as the traditional multilayer perceptron (MLP) does not scale well and suffers from the curse of dimensionality due to the full connectivity between nodes. Convolutional layers apply filters to that convolve the input image to identify simple to complex patterns and pooling layers helps to reduce the dimensions/features of the output (feature maps) from the convolutional layers to make said feature maps more robust.

Research Design

For this assignment, we will look at 2000 cat and dog images (1000 each) and we will use a 2x2 factorial design (DNN vs CNN, Grayscale vs RGB) to experiment with image classification. The output of the classification should be assessing how well the neural network is able to detect whether the subject in the photo is a cat or dog. Evaluation of the experiment will be conducted using train/test accuracy, as well as model runtime.

Technical Overview

In order to conduct the classification experiment, we will carry out a factorial design of two levels on two experimental factors, as stated earlier. We also will be making use of deep neural networks (2 hidden layers) and convolutional neural networks (3 convolutional layers, two pooling layers, and two fully connected layers). Four models will be compared against each other: Model A – Grayscale DNN, Model B – RGB DNN, Model C – RGB CNN, Model D – Grayscale CNN. For classification, we will compute the softmax cross entropy between logits and the model will be optimized using the Adam Optimization Algorithm (similar to the classical

stochastic gradient descent) because it is computationally efficient. For the DNNs, we used a batch size of 100 and 50 epochs. For the CNNs, we used a batch size of 256 and 100 epochs. The experiment will be conducted using a Python API for TensorFlow, which is an open source machine learning library, initially developed by Google, and TFLearn, a deep learning library. The metrics we will be comparing will be the model performance as well as model runtime (in seconds).

Results

	Train Accuracy	Test Accuracy	Processing Time (seconds)
Model A	0.8700	0.5650	5.050180
Model B	0.8700	0.5650	4.909892
Model C	0.9466	0.6860	903.163326
Model D	0.9387	0.6525	935.644844

For our benchmark model, we started off with a deep neural network with two hidden layers, considering 64x64 grayscale images. For the following models we experimented with changing the class of neural networks as well as the colors of the images (grayscale vs. RGB). Although, it took a long time in terms of computational runtime (903 seconds), Model C (CNN, RGB) yielded the highest training and test accuracy.

When concerned with achieving the highest possible accuracy in image classification, CNNs would be the best ML model to use. Within a CNN, we want to use convolutional, pooling, and fully connected layers to yield the best results. Based off of this experiment, RGB images with high contrast work the best.