**ADVANCED MACHINE LEARNING ASSIGNMENT-2**

Convolution Report

Supriya Mattapelly

**Introduction:**

Using a portion of the popular "Dogs-vs-Cats" dataset extracting from Kaggle to working on google collab gives me a challenging chance to create a very successful model with a small amount of data. Convolutional neural networks, which are also referred to as convnets, are the preferred method in computer vision for image identification, object detection, and segmentation tasks because of its remarkable capacity to learn and identify spatial patterns in images. Even with the restricted amount of data provided, I am confident that I can get remarkable outcomes by employing convolutional neural networks' capacity to extract and recognize crucial features from images.

Using a small dataset, I would train my model, improve it with state-of-the-art transfer learning methods, and test its performance with suitable assessment criteria. With minimum input, I intend to create an accurate and efficient convolutional neural network that can classify images from the "Dogs-vs-Cats" dataset. Demonstrating the potential of my model stimulates me, and I am driven to explore new frontiers in computer vision with limited data. My convolutional neural network will undoubtedly contribute significantly to the field of computer vision because it is focused on efficiency and creativity.

**Problem:**

The Cats-vs-Dogs dataset binary classification task aims to determine whether an image belongs in the dog or cat category.

**Pre-trained Model:**

The original dataset is huge and diverse, and its properties make it useful for a variety of computer vision applications. A previously trained network can be utilized as a generic model. The capacity of deep learning to transfer learned properties across tasks is one of its main advantages over other machine learning techniques.

Consider a huge convolutional neural network that was trained on the 1,000 unique classes and 1.4 million annotated images in the ImageNet dataset. Numerous animal categories, including various cat and dog breeds, are included in this dataset. The architecture of this network, referred to as VGG16, is a fundamental and widely utilized convnet architecture for ImageNet.

**Data Implementation:**

We suggest using data augmentation techniques to raise our model's accuracy. Even with limited datasets, we can obtain good results by randomly modifying the provided training samples to create new data. Consequently, this helps with generalization since the model will never see the same image twice while it is being trained.

We suggest using data augmentation techniques to raise our model's accuracy. Even with limited datasets, we can obtain good results by randomly modifying the provided training samples to create new data. Consequently, this helps with generalization since the model will never see the same image twice while it is being trained.

**Methods:**

This is a binary classification problem using the Cats-vs-Dogs dataset, where you have to guess if a photo is in the dog or cat class.   
• Get started with the image files.   
• Convert the JPEG content to pixel grids in RBG.   
• Transform them into floating-point tensors.   
The pixel values (ranging from 0 to 255) should be modified to the [0, 1] interval since, as you may know, neural networks prefer small input values.

The 543MB Cats-vs-Dogs dataset includes 25,000 images of dogs and cats, with 12,500 images in each class (condensed). We'll create a new dataset with three subsets after downloading and unzipping it: a training set with 1000 samples for each class, a validation set with 500 samples for each class, and a test set with 500 samples for each class. Because the problem we're working on is more complex and has a larger image, we need to increase the neural network's capability. We plan to accomplish this by including a new stage into our current Conv2D + MaxPooling2D design.

This will ensure that the feature maps are not overly large when we reach the Flatten layer by increasing network bandwidth while concurrently decreasing their size. The original dimensions of our input photographs are 150x150. As we move through the network layers, the feature maps get smaller and smaller until they are 7x7 right before the Flatten layer. Although the input size selection is a bit random, it makes sense in this instance.

**Table for Model From Scratch**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model number** | **Train Size** | **Validation and Test Sample Size** | **Data Augmentation** | **Test Accuracy %** | **Validation Accuracy %** |
| Model 1 | 1000 | 500,500 | NO | 76.8 | 70.6 |
| Model 1a | 1000 | 500,500 | YES | 67.1 | 64.2 |
| Model 2 | 1500 | 500,500 | NO | 83 | 71.9 |
| Model 2a | 1500 | 500,500 | YES | 70.37 | 70.3 |
| Model 2b | 1500 | 500,500 | YES | 81.7 | 73.2 |
| Model 2c | 1500 | 500,500 | NO | 72.7 | 73.8 |

**Table for Pre-Trained Models**

|  |  |  |
| --- | --- | --- |
| **Data Augmentation** | **Train Accuracy %** | **Validation Accuracy %** |
| NO | 99.6 | 97 |
| YES | 95.8 | 97.2 |

**Summary:**

The tables above contain the model settings and the sample sizes for the train, test, and validation sets.We present findings for the model trained from scratch, with and without data augmentation, and for models trained with different train and validation sizes. For the pre-trained model, we compare the accuracy, validation accuracy and data augmentation.

The findings show that models trained with and without data augmentation did not consistently perform better than those trained without. The accuracy of the model is further improved by expanding the training set or changing the validation set's size. Data augmentation had no effect on the model's accuracy or validation accuracy when we compared the pre-trained model with and without data augmentation. All things considered, pre-trained models perform better than models built from scratch, particularly in situations when there is a dearth of training data.