Assignment 2

Convolution Report

*Manasa Akkinapally*

**Introduction :-**

I have a challenging opportunity to create a highly successful model in google colab with limited data by extracting data from Kaggle and downloading the data sets to drive, which use a subset of the popular "Dogs-vs-Cats" dataset. The industry standard for image identification, object detection, and segmentation tasks in computer vision is convolutional neural networks, or convnets for short. Convnets are renowned for their extraordinary capacity to learn and recognize spatial patterns in images. I'm confident that by using convolutional neural networks' capacity to extract and recognize key features from images, I will be able to get outstanding results even with the restricted amount of data provided.

My ideal workflow would be to use a small dataset for training, advanced transfer learning methods for fine-tuning, and suitable assessment criteria for assessing the model's performance. My goal is to create a convolutional neural network that is both accurate and efficient, capable of accurately classifying images from the "Dogs-vs-Cats" dataset with minimal input. I'm driven to push the limits of what is feasible in computer vision with little data, and I'm thrilled to demonstrate the possibilities of my model. I am confident that my convolutional neural network will significantly advance the field of computer vision because it focuses on efficiency and innovation.

Pre-trained model :-

A pretrained network can be utilized as a general model due to its properties, which can be used to a variety of computer vision applications. The original dataset is huge and diversified. Transferring learned characteristics from one task to another is one of deep learning's main advantages over other machine learning techniques.

Let us consider a massive convolutional neural network trained on the 1,000 different class and 1.4 million annotated photographs of the ImageNet dataset. Animal classifications from various breeds of dogs and cats are included in this collection. The basic convnet architecture used for ImageNet is called VGG16, and it is the design of this network.

Data Augmentation:

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.

Techniques :-

To achieve our specific goal, we wish to randomly apply effects like flipping, rotating, and zooming to the training set of photographs. By doing this, we might produce variations of the existing images, adding diversity to the dataset and enhancing our model's resilience.

* We have to determine if the image in the Cats-vs-Dogs dataset belongs to the dog or cat class. This is a binary classification problem.
* Let the picture files open.
* Transform the JPEG content into RBG pixel grids.
* Use floating point to transform them into tensors.
* Due to the fact that you may know, neural networks prefer small input values, rescale the pixel values (between 0 and 255) to the [0, 1] interval.

The 543MB Cats-vs-Dogs dataset includes 25,000 images of dogs and cats, with 12,500 images in each class. (compressed). 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. By doing this, we can ensure that the feature maps won't be overly large when we get to the Flatten layer while also increasing network bandwidth. The feature maps in our input photos start off as 150x150 images and gradually get smaller as we move through the network layers, reaching a final size of 7x7 right before the Flatten layer. Though rather arbitrary, this input size selection makes sense in this specific scenario.

**Table for Model from Scratch**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model no** | **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 |

**Conclusion:**

* + From the observations, we can say that the choice of network and change of training samples has great impact on the accuracy, which tells that there are dependable.
  + choice of network and training samples are both critical factors in determining the performance of a CNN. The selection of the appropriate structure and training data requires careful

consideration and experimentation to achieve optimal results.

* + The structure of CNN is determined by choice of network, including activation functions, size, and arrangement of layers and also depends on complexity and training data available.
  + The choice of training samples, on the other hand, determines the specific examples on which the network is trained. Here, in image classification, they as labelled as cat, dog, cat, cat etc.,
  + CNN may overfit, if the training sample is low and performs poorly on unseen data and if the training sample is high, it may become too noisy and struggles to learn patterns. We are using different optimization techniques such as drop out and data augmentation methods to improve accuracy and tune model not to overfit.
  + For a scratch model, training samples must be increased to get best accuracy that to a very less accuracy compared to pretrained model.
  + But for a pretrained model, it learns important features for the data set and reduce the amount of training data to achieve good performance on a new task. For large datasets, pretrained model is highly acceptable because it highly generalizes by training on large classes.
  + We can observe that from the results, that pretrain require less training sample but for scratch model we need to increase the training samples . For a scratch model , training samples must be increased to get best accuracy that to a very less accuracy compared to pretrained model.
  + Additionally, we can observe that the test accuracy has been decreasing in scratch model when it reaches to certain level, which means that the model is too complex for that model, whereas pretrained model accuracy has been increasing as the sample size increasing.

Overall, pre-trained models often have a higher accuracy compared to scratch models because they have been trained on larger and more diverse datasets, have more complex architectures, are optimized using advanced techniques. However, scratch models can still be effective if they are trained properly and have a suitable architecture for the task at hand.