Assignment 3

Convolution Report

*Addula Nikhil Reddy*

***Introduction :-***

*Working on Kaggle using a subset of the well-known "Dogs-vs-Cats" dataset provides me a difficult opportunity to build a highly effective model with limited data.Convolutional neural networks, or convnets, are well-known for their exceptional ability to learn and recognize spatial patterns in images, making them the go-to choice in computer vision for image recognition, object detection, and segmentation tasks. Despite the limited data supplied, I am certain that I will be able to get amazing results by utilizing the ability of convnets to extract and identify essential attributes from photographs.*

*I'd want to train my model on a limited dataset, fine-tune it using cutting-edge transfer learning techniques, then evaluate its performance using appropriate assessment criteria. I am committed to developing an accurate and efficient convolutional neural network that can correctly categorize photos from the "Dogs-vs-Cats" dataset with a small amount of input. I am excited to show off the capabilities of my model and am motivated to push the boundaries of what is possible in computer vision with minimal data. By concentrating on innovation and efficiency, I am certain that my convolutional neural network will make a significant contribution to the area of computer vision.*

***Pre-trained model :-***

*The original dataset is both large and diverse, a pretrained network may be used as a generic model, with its characteristics applicable to a wide range of computer vision applications. One of the primary advantages of deep learning over other machine learning approaches is its ability to transfer learnt characteristics across tasks.*

*Consider a huge convolutional neural network trained on the ImageNet dataset, which includes 1.4 million annotated pictures and 1,000 distinct classes. This dataset contains multiple animal classifications, including several cat and dog breeds. This network's design is known as VGG16, and it is a basic and extensively used convnet architecture for ImageNet.*

***Data Augmentation:***

*We propose to apply data augmentation approaches to improve the accuracy of our model. By producing new data from the given training samples using random modifications, we may acquire good results even with restricted datasets. As a consequence, the model will never encounter the same picture twice during training, which aids with generalization.*

*For our specific objective, we want to perform modifications to the photos in the training set at random, such as flipping, rotating, and zooming. By doing so, we may generate variants of the current photos, increasing the dataset's variety and boosting the resilience of our model.*

***Techniques :-***

*The Cats-vs-Dogs dataset is a binary classification issue in which you must estimate whether a picture belongs to the dog or cat class.*

*• Open the image files.*

*• Convert the JPEG material to RBG pixel grids.*

*• Convert them to tensors in floating point.*

*• Rescale the pixel values (between 0 and 255) to the [0, 1] interval (neural networks, as you know, like tiny input values).*

*The Cats-vs-Dogs dataset comprises 25,000 photos of dogs and cats (12,500 from each class) and is 543MB in size. (compressed). After downloading and uncompressing it, we will construct a new dataset with three subsets: a training set with 1000 samples of each class, a validation set with 500 samples of each class, and lastly a test set with 500 samples of each class. We need to expand the capacity of our neural network due to the higher picture size and more intricate nature of the challenge we're working on. To do this, we will add a new stage to our existing Conv2D + MaxPooling2D architecture.* *This will increase network capacity while simultaneously lowering the size of the feature maps, ensuring that they are not too huge when we reach the Flatten layer. Our input photos are originally 150x150 in size, and as we proceed through the network layers, the feature maps steadily shrink in size until they are 7x7 just before the Flatten layer. This choice of input size is rather random, but it is appropriate for the particular situation.*

***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:***

*The model settings, as well as the sample sizes for the train, test, and validation sets, are provided in the tables above. We include results with and without data augmentation for the model from scratch, as well as models trained with an increase in train size or with varied train and validation sizes. We compare the accuracy, validation accuracy, and data augmentation for the pre-trained model.*

*Based on the results, we can see that models trained with data augmentation did not consistently outperform those trained without it. Increasing the size of the training set or altering the size of the validation set also enhances the model's accuracy. When we compare the pre-trained model with and without data augmentation, we can find that data augmentation did not increase the model's accuracy or validation accuracy. Overall, pre-trained models outperform models created from scratch, especially when little training data is available.*