AML ASSIGNMENT-2

MEGHANA THOUTAM (811298610)

Summary:

Our project aims to develop a convolutional neural network (convnet) specifically for computer vision tasks using Kaggle’s "Dog-vs-Cats" dataset, which presents a challenge due to its small size. Convnets are highly effective in computer vision because they excel at detecting spatial patterns in images, making them ideal for tasks like image classification, segmentation, and object detection.

Even with the limited data available, we believe our convnet will yield promising results. Convnets are known for their ability to generalize effectively from small datasets by identifying critical image features. Our strategy is to train the model on the data at hand, improve it with transfer learning, and evaluate its performance using key metrics. Our objective is to create a convnet that accurately classifies images in the "Dog-vs-Cats" dataset with minimal data usage.

Problem:

The goal of the Cats-vs-Dogs dataset classification task is to determine if an image contains a dog or a cat.

Techniques and Dataset:

The Cats-vs-Dogs dataset includes 25,000 images (split equally between dogs and cats). We plan to structure the data into three subsets: a training set with 1,000 samples per class, a validation set with 500 samples per class, and a test set with 500 samples per class. All data will be downloaded and uncompressed.

We are expanding our neural network's capacity to handle the increased complexity of this task. To achieve this, we are adding another layer to our Conv2D + MaxPooling2D model. This adjustment helps control the size of feature maps before reaching the Flatten layer and enhances the network’s overall capacity. The input images start at 150x150 pixels, and as they move through the network layers, the feature maps reduce in size to 7x7 before reaching the Flatten layer, which is well-suited for this task.

AML ASSIGNMENT-2
MEGHANA THOUTAM (811298610)

Preprocessing:

For preprocessing:

- Load the image files.
- Convert JPEG images into RGB pixel grids and transform them into floating-point tensors.
- Normalize pixel values from 0-255 to a range of [0, 1], as neural networks perform better with smaller input values.

Data Augmentation:

To enhance our model's accuracy, we will apply data augmentation techniques. These techniques generate new variations of training images, allowing the model to generalize better even with a limited dataset. We plan to introduce transformations like random flips, rotations, and zooms to the training images, creating a more diverse dataset and improving the model's robustness.

Pretrained Model:

Since the dataset contains various dog and cat breeds, we plan to use a pretrained model, such as VGG16, which is a popular convnet designed for ImageNet. Pretrained models are effective for a wide range of computer vision tasks because they are trained on large, diverse datasets like ImageNet, which contains 1.4 million labeled images across 1,000 classes.

We will focus on two strategies for utilizing the pretrained network: feature extraction and fine-tuning. Initially, we will perform feature extraction without data augmentation, and later incorporate data augmentation to improve the model's performance.

Q1. Consider the Cats & Dogs example. Start initially with a training sample of 1000, a validation sample of 500, and a test sample of 500 (like in the text). Use any technique to reduce overfitting and improve performance in developing a network that you train from scratch. What performance did you achieve?

A training set of 1,000 images (with 500 for validation and 500 for testing) was used from the Cats & Dogs dataset. To counteract the risk of overfitting with such a small sample size, I implemented a 50% dropout strategy.

AML ASSIGNMENT-2
MEGHANA THOUTAM (811298610)

Hyperparameter tuning: The batch size was configured to 255, and data transformation was done using the flattening technique. The model's test accuracy reached 72.22%, while the validation accuracy was 72.7%.

Q2. Increase your training sample size. You may pick any amount. Keep the validation and test samples the same as above. Optimize your network (again training from scratch). What performance did you achieve?

The results are as follows:

- Validation accuracy: 71.10%
- Test accuracy: 73.4%

These findings indicate an improvement over previous results (Question 1) for several reasons:

The model's performance has benefited from the increase in our training sample size to 500 (from 1,000 to 1,500). This change has led to over a 10% increase in both training and validation accuracy. Furthermore, we incorporated data augmentation alongside the convolutional layers, which enhanced feature extraction and contributed to better overall performance.

Q3. Now change your training sample so that you achieve better performance than those from Steps 1 and 2. This sample size may be larger, or smaller than those in the previous steps. The objective is to find the ideal training sample size to get best prediction results.

Increasing the amount of training data is a well-recognized method for improving model performance, but identifying the optimal sample size can be difficult. In this instance, employing data augmentation techniques and adding 500 additional samples led to a noticeable improvement in model performance, increasing accuracy from 68.8% to 70.1%.

However, despite the larger sample size and improved data within the specified convolutional architecture, the model demonstrates limited capacity to learn new information, highlighting this issue clearly.

This finding indicates that there may be a need to investigate alternative strategies to further optimize the model's performance.

AML ASSIGNMENT-2
MEGHANA THOUTAM (811298610)

Q4. Repeat Steps 1-3, but now using a pretrained network. The sample sizes you use in Steps 2 and 3 for the pretrained network may be the same or different from those using the network where you trained from scratch. Again, use any and all optimization techniques to get best performance.

When training from scratch, it is essential to utilize all available optimization techniques to achieve the best possible performance.

Pre-Trained Model without Augmentation:

The model reached a validation accuracy of 98.0% and a test accuracy of 96.8%. While the test accuracy is promising compared to the initial training results of a smaller model, there is a concerning indication of overfitting.

This overfitting is apparent even with the use of dropout regularization at a relatively high rate, as demonstrated in the accompanying plots. Although the model performs well on the validation dataset—used for fine-tuning hyperparameters—it may struggle to generalize effectively to unseen data. The dropout plots suggest that overfitting begins early in the training process.

Pre-Trained Model with Data Augmentation:

Selecting the appropriate data for evaluating a model is essential. Strong performance on one dataset may not be indicative of results on other datasets, particularly when considering the varying levels of complexity across different datasets.

This is evident in the comparison of the pre-trained model's accuracy: it achieved 98.6% without data augmentation and 97.7% with data augmentation.

The performance of different models based on the number of training samples is outlined as follows:

- **Model 1**, which used 1,000 training samples, achieved a validation accuracy of 72.1% and a test accuracy of 72.7%.
- **Model 2**, trained on 1,500 samples, recorded a validation accuracy of 71.1% and a test accuracy of 73.4%.
- **Model 3** utilized 2,000 training samples, resulting in a validation accuracy of 68.8% and a test accuracy of 70.1%.

- **Model 4**, a pre-trained model without data augmentation, displayed significantly higher performance, with a validation accuracy of 98.0% and a test accuracy of 96.8%.
- When data augmentation was applied to **Model 4**, the validation accuracy rose to 98.6%, and the test accuracy improved to 97.7%.

Conclusion:

This report examines the impact of training data size, validation set size, and data augmentation techniques on the performance of both scratch-built and pre-trained models. The key findings are as follows:

- Enhancing the training data or modifying the validation set size can lead to improved accuracy for both types of models.
- Data augmentation did not yield significant accuracy gains for either model.
- Pre-trained models typically outperform scratch models, particularly when data is limited, as they benefit from knowledge gained from prior tasks.