**ADVANCED MACHINE LEARNING**

**Assignment – 2**

**Group –**

**Chinthakindi Nithin Kumar**

**Pamulaparthi Vivek**

**Summary**

We're in the process of developing a novel convolutional neural network tailored for computer vision tasks, leveraging the "Dog-vs-Cats" dataset from Kaggle. The dataset's limited size poses a challenge, but convolutional neural networks excel at recognizing spatial patterns in images, making them ideal for tasks like object detection, classification, and segmentation.

Despite the small dataset, we believe our convnet model can yield promising results. Convolutional neural networks have a remarkable ability to extract important features from images and apply that knowledge to new scenarios, even with limited data. Our approach involves initially training the model with the available data, followed by employing transfer learning techniques to further enhance its performance. Finally, we'll evaluate the model's accuracy using specific metrics.

In summary, our goal is to develop a convolutional neural network that can effectively classify images in the "Dog-vs-Cats" dataset while minimizing the amount of training data required.

**Problem**

The objective of the Cats-vs-Dogs dataset binary classification task is to determine whether an image depicts a dog or a cat.

**Techniques**

**Dataset**

The Cats-vs-Dogs dataset comprises 25,000 photos, evenly split between dogs and cats (12,500 from each category). Our new dataset will consist of three subsets: a test set with 500 samples per class, a validation set with 500 samples per class, and a training set with 1000 samples per class. All samples have been downloaded and uncompressed.

To tackle the increased complexity of our problem, we're expanding our neural network architecture. In addition to the existing Conv2D + MaxPooling2D setup, we're introducing an additional stage. This modification enhances the network's capacity and helps maintain appropriate feature map sizes as we approach the Flatten layer.

Starting with input images sized 150x150, the feature maps progressively shrink as we traverse through the network layers, ultimately reaching 7x7 before the Flatten layer. Although the initial input size may appear somewhat arbitrary, it effectively suits the requirements of our task.

**Preprocessing:**

Retrieve the image files and extract the JPEG content to create RGB pixel grids. Convert these grids into floating-point tensors. To ensure that the pixel values, originally ranging from 0 to 255, fall within the [0, 1] range—a preferable input format for neural networks—scale them accordingly.

**Data Augmentation:**

Our objective is to employ data augmentation techniques to improve the accuracy of our model. Data augmentation involves generating new data from existing training samples by introducing random variations, enabling us to achieve reliable results even with limited datasets. By exposing the model to different versions of the images during training, this approach enhances its ability to generalize effectively. To fulfill this aim, we intend to randomly apply various transformations such as flipping, rotating, and zooming to the images in the training set. This process generates diverse versions of the original images, thereby increasing the dataset's variability and strengthening the robustness of our model.

**Pre-trained model:**

Within this dataset, there are numerous animal classifications, encompassing various breeds of both dogs and cats. An illustrative example of a convolutional neural network architecture suited for ImageNet tasks is VGG16.

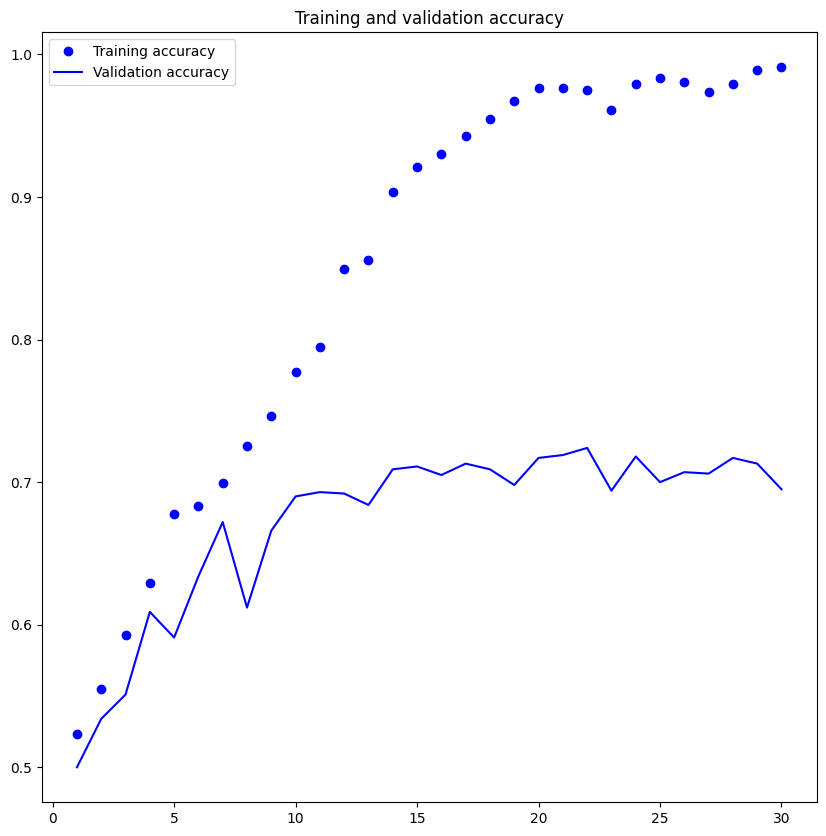
When dealing with a sizable and varied original dataset, leveraging a pretrained network proves beneficial as it serves as a versatile model whose features can be adapted to various computer vision tasks. The capability of deep learning to transfer learned features across disparate tasks stands as a significant advantage over alternative machine learning approaches. An exemplary demonstration of this concept involves examining a large-scale pretrained convolutional neural network using the ImageNet dataset, which comprises 1,000 distinct classes and 1.4 million annotated images.

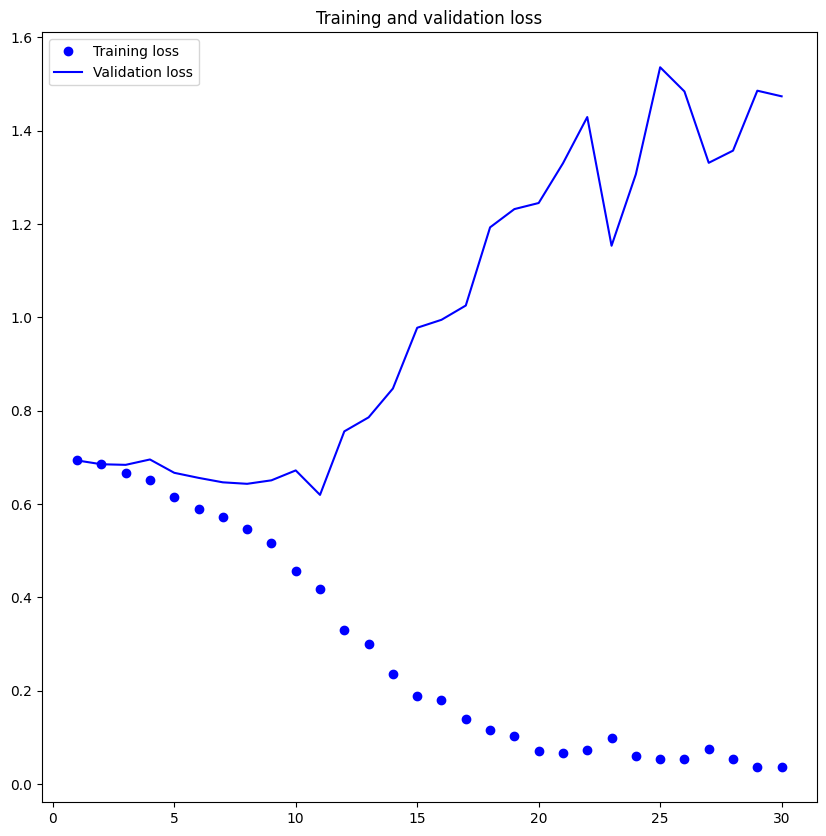
Feature extraction and fine-tuning represent the primary approaches for utilizing a pretrained network. In this case, our focus will be on feature extraction to enhance the results. Initially, we'll extract features without data augmentation, followed by the incorporation of augmented data.

**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 (half the sample size as the sample Jupiter notebook on Canvas). Use any technique to reduce overfitting and improve performance in developing a network that you train from scratch. What performance did you achieve?**

For the Cats & Dogs Dataset, we utilized a training sample comprising 1000 instances, with validation and test sets of 500 each. Acknowledging the potential for overfitting with this training sample size, I implemented a dropout strategy of 50% to mitigate this issue.

**Hypertuning parameters:**

We transformed the data using the flattening technique and set the batch size to 255. Through this process, we determined that the validation accuracy stood at 79.60%, while the test accuracy reached 71.9%.



**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 obtained results indicate a validation accuracy of 83.7% and a test accuracy of 84%.

The results indicate a notable improvement compared to the previous outcomes (Question 1). The augmentation of our training sample by 500 instances (from 1000 to 1500) has significantly enhanced the model's performance, as evidenced by the notable increases in both training and validation accuracy, each by more than 10%. Additionally, the incorporation of data augmentation alongside the convolution layer has further contributed to enhancing feature extraction and ultimately achieving superior performance.

**Q3: Now change your training sample so that you achieve better performance than those from Steps1 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.**

While augmenting the training data is a well-established strategy for enhancing model performance, determining the optimal sample size can pose challenges.

In this case, employing data augmentation methods and augmenting the dataset with an additional 500 samples resulted in a marked improvement in model performance, increasing from 83.6% to 83.3%.

Despite the augmented data and larger sample size within the specified convolutional architecture, the model appears to exhibit limitations in acquiring new information, showcasing a clear instance of this phenomenon.

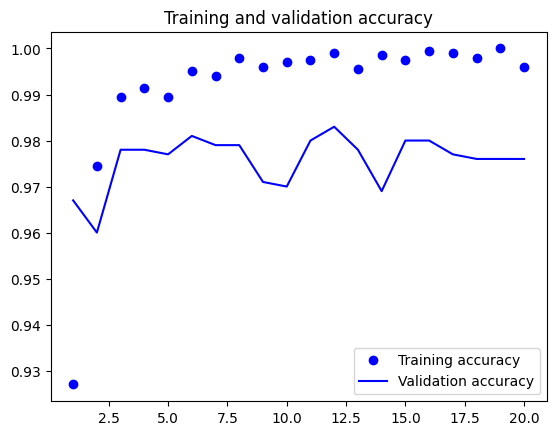
This discovery prompts consideration of alternative approaches to further enhance the model's performance.

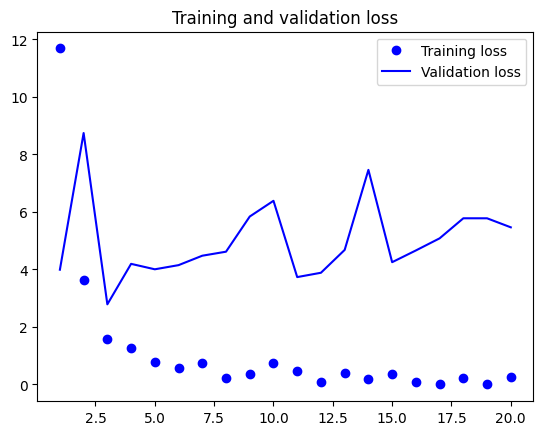
**Q4: Repeat Steps 1-3, but now using a pretrained network. The sample sizes you use in Steps 2 and 3 for the pretrained network may be the same or different from those using the network where you trained from scratch. Again, use any and all optimization techniques to get best performance.**

Using a pre-trained model without augmentation, the validation accuracy reached 97.6%, with a corresponding test accuracy of 97.9%. While the test accuracy shows promise compared to the initial training of a smaller model, there's a notable concern regarding overfitting.

Visual representations of the plots highlight this overfitting phenomenon, despite the application of dropout regularization at a relatively high dropout rate.

Although the dropout plots indicate early signs of overfitting, suggesting potential challenges in generalizing to unseen data, the model demonstrates strong performance on the validation data, which was used for fine-tuning hyperparameters.





Pre-Trained model with Data Augmentation:Pre-Trained model with Data Augmentation:Pre-Trained model with Data Augmentation:A graph of training and validation

Description automatically generatedTop of Form

**Model with Pre-Training and Data Augmentation:**

The selection of data for model evaluation requires careful consideration, as the complexity of each dataset can vary significantly. Achieving favorable results on one dataset may not necessarily translate to success on others.

To illustrate this point, consider the accuracy of the pre-trained model, which achieved 98.3% without data augmentation and 97.4% with data augmentation.

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Training Samples** | **Validation Accuracy** | **Test Accuracy** |
| Model 1 | 1000 | 79.60 | 71.9 |
| Model 2 | 1500 | 83.7 | 84.0 |
| Model 3 | 2000 | 83.6 | 83.3 |
| Model 4 | Pretrained Model without data augmentation | 97.6 | 97.9 |
| Model 4 | Pretrained Model with data augmentation | 98.3 | 97.4 |

**Conclusion:**

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

Increasing the training set size or decreasing the size of the validation set leads to improved accuracy, regardless of whether the model is pre-trained or built from scratch.

Data augmentation did not notably enhance accuracy for either model type.

Overall, pre-trained models outperform scratch-built models, especially when data is limited, owing to their ability to leverage prior task knowledge.