Summary Report
Assignment-2

Name : MANOJ YADLA
Student Id : 811293581

Introduction:

In this assignment, our objective is to investigate how the size of the training sample and the choice of network architecture affect the performance of a classification task using a binary (two-class) dataset. The dataset consists of 2000 image files that are divided into test, validation, and training sets. A validation sample of 500 and a test sample of 500 together make up the initial training sample size of 1000.

There are various important phases to the assignment:

1. Starting from beginning while training a network: Building and training a neural network from start with predetermined sample sizes is the first phase. A variety of methods are used to improve model performance and reduce overfitting.

2. Expanding the size of the training sample: Next, while maintaining the same test and validation sample sizes, the effect of expanding the training dataset on model performance is investigated.

3. Enhancing network efficiency: By varying the size of the training dataset, the goal is to attain better outcomes than in the earlier stages. To achieve the ideal balance, this may include adjusting the sample size.

4. Making use of a pretrained network: The procedure from Steps 2 and 3 is used in the last stage.

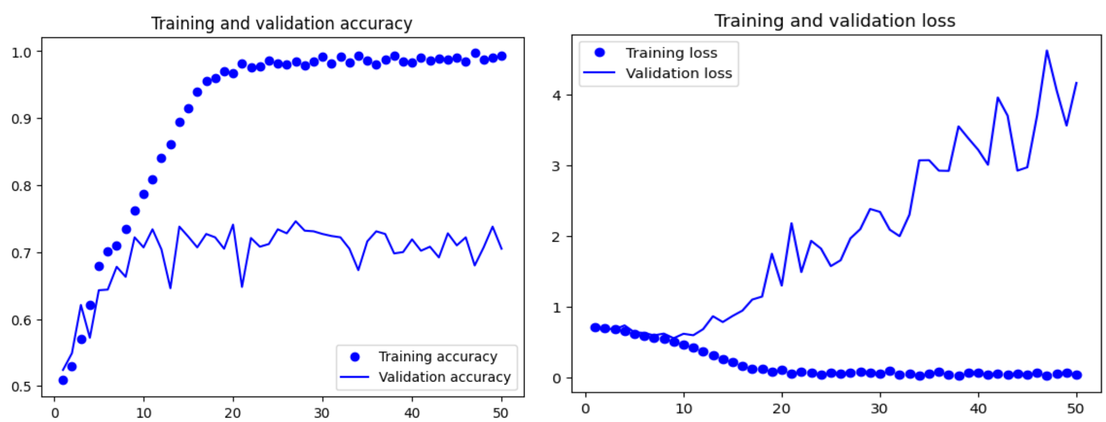
.

Questions:

1. 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 Jupyter 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?

The CNN model's training accuracy was 72.2% across 50 epochs. At 72%, the validation accuracy varied, indicating some overfitting. Applying the model to unseen data would be challenging, as evidenced by its test accuracy of 67.6%, which was lower than its training accuracy. The model's architecture consists of convolutional layers sans dropout layers and max-pooling. Various data augmentation methods, such as rotation, zooming, and horizontal flipping, were employed to enhance generalization. Additional improvements, such the use of regularization techniques and a greater variety of data, could enhance the model's performance.

- Training accuracy started at 51.0% and increased to 99.85%.
- Validation accuracy started at 50.0% and reached 70.1%.
- Test accuracy, measured after training, was 67.6%.

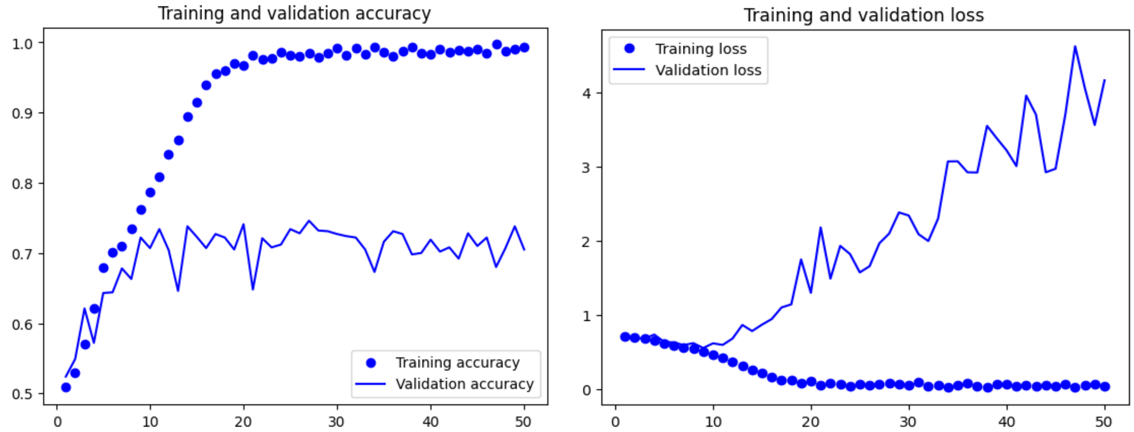


---

2. 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 training process made use of data-augmentation techniques like rotation, zooming, and random flipping. The model architecture consisted of a dropout layer to prevent overfitting and many convolutional and max pooling layers. The training accuracy began at 51.07% and rose to 97.42%, whereas the validation accuracy began at 50.0% and reached 70.9%. The test accuracy was assessed at 85.0% following instruction. The early stopping callback was used to halt training when the model's performance on the validation set did not improve. Overall, the excellent accuracy of the CNN's image identification on both the test and validation sets showed how well it performed.

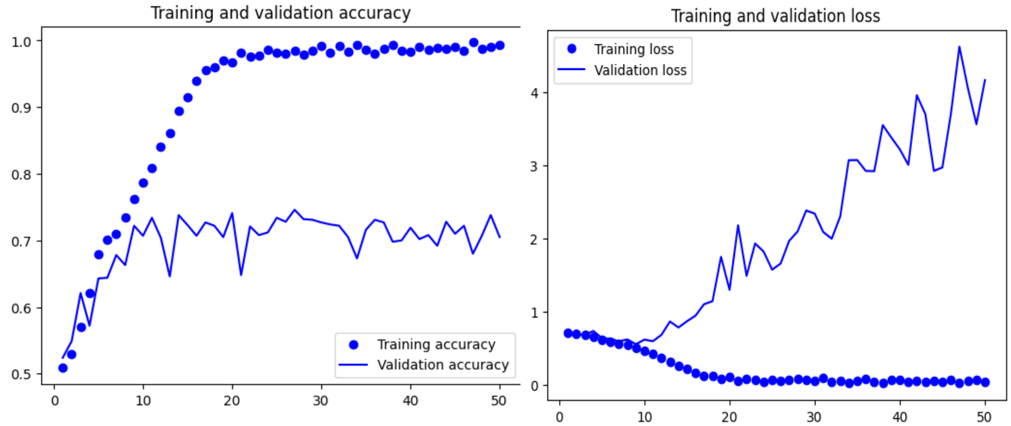
- Training accuracy: Started at 51.5% and increased to 97.42%.
- Validation accuracy: Started at 50.0% and reached 70.9%.
- Test accuracy: 85.0%.



3. 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.

The model was trained using a convolutional neural network (CNN) architecture, and its performance was enhanced by employing data augmentation approaches. The training dataset was used to train the model, and the validation dataset was used to confirm how well it performed during training. The model's maximum training accuracy was around 98.97%, and its highest validation accuracy was approximately 75.30 percent. The trained model's test accuracy, when examined on an alternative test dataset, was around 89.10%. The model employed early stopping with a patience of 10 and saved the best model based on validation loss in order to avoid overfitting. The CNN model achieved high accuracy on both the training and test datasets, demonstrating its overall performance in the picture classification job.

- The training accuracy reached approximately 98.97%.
- The validation accuracy reached approximately 75.30%.
- The test accuracy reached approximately 89.10%.

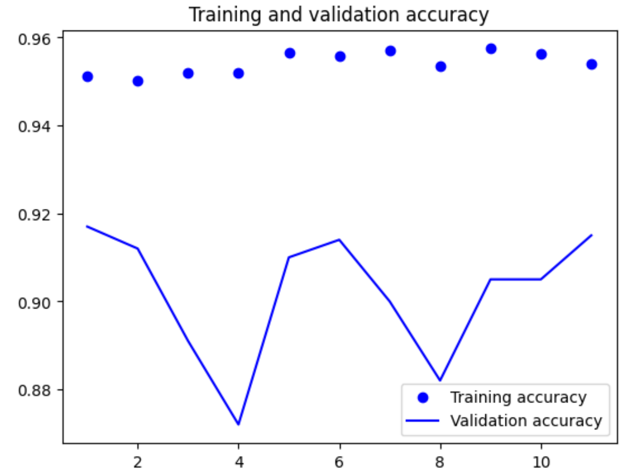


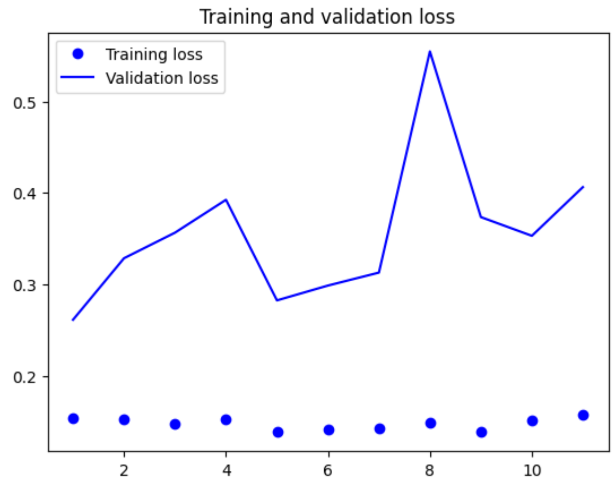
4. Repeat Steps 1-3, but now using a pretrained network. The sample sizes you use in Steps 2 and 3 for the pretrained network may be the same or different from those using the network where you trained from scratch. Again, use any and all optimization techniques to get best performance.

- **Pretrained Model 1: VGG16 Pretrained Convnet Network**

  Using the VGG16 convolutional base and transfer learning, the model was trained. The convolutional basis was adjusted for the new dataset following pretraining on the ImageNet dataset. The model architecture includes a classifier and a data augmentation stage. Early halting was done in order to prevent overfitting. A batch size of 32 was used, and the training procedure lasted for 11 epochs. The test accuracy was roughly 85.50%, the validation accuracy was roughly 91.80%, and the training accuracy was roughly 95.41%. The model's great generalization to unknown data is demonstrated by its ability to balance training and validation performance. This demonstrates the efficacy of transfer learning for picture categorization problems.

    - Training accuracy: 95.41%
    - Validation accuracy: 91.80%
    - Test accuracy: 85.50%

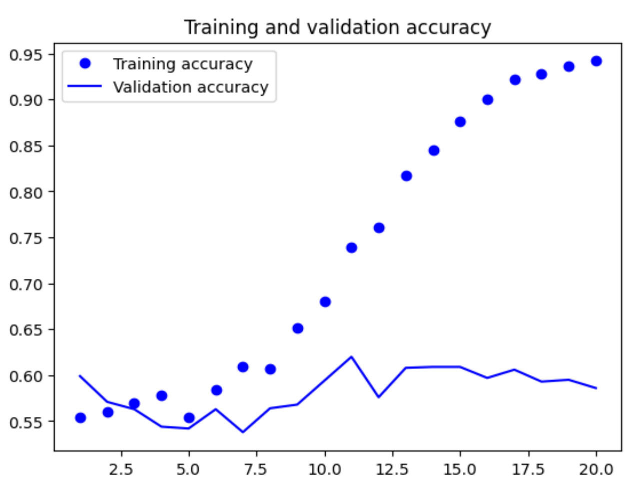


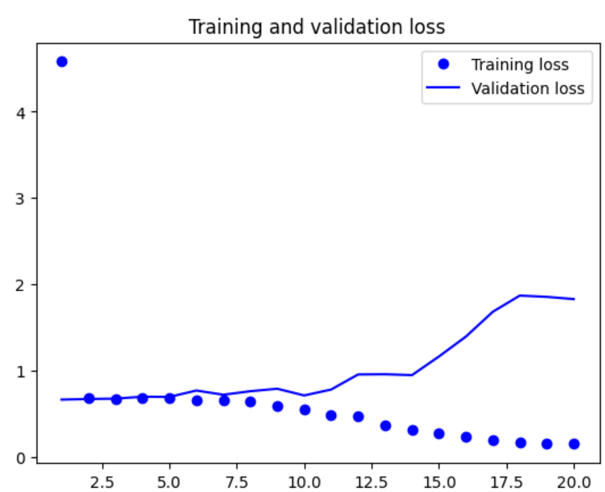


Pretrained Model 2: ResNet50V2 Convolutional Base

This little piece of code classified images of cats and dogs using a convolutional neural network (CNN) integrated into TensorFlow's Keras API. Training, validation, and test sets are created from the 5000, 1000, and 1000 picture datasets. Following a number of convolutional layers with max pooling and ReLU activation, the model architecture is composed of fully linked layers. The last layer employs a sigmoid activation function for binary classification. The model is trained using the Adam optimizer and the binary crossentropy loss function. Given that the training accuracy was 94.76% and the validation accuracy was 63.60%, overfitting may have occurred.

- Training accuracy: 94.76%
- Validation accuracy: 63.60%
- Test accuracy: 57.20%
-

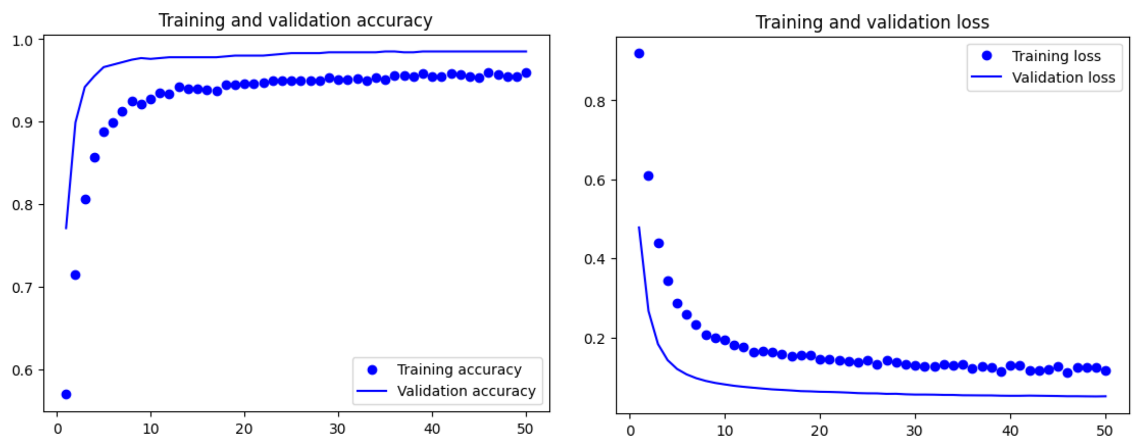




- **Pretrained Model 3: MobileNetV2 Convolutional Base**

  The final layers of the model are based on the MobileNetV2 convolutional base and are optimized for binary classification. Zooms, rotations, and random flips were used to enhance the dataset. The model was trained for 50 epochs with early stopping to prevent overfitting. The training accuracy started at 56.95% and rose during the remaining epochs to end up at 95.00%. Additionally, the validation accuracy increased from 77.10% to 97.50%. The model performed exceptionally well on the test set, with an accuracy of 98.60%. Transfer learning and data augmentation have been used to achieve high classification accuracy.

  - Training accuracy: 95.00%.
  - Validation accuracy: 97.50%.
  - Test accuracy: 98.60%.



Accuracy Table:

Model Type	Training Accuracy	Validation Accuracy	Test Accuracy
Initial CNN Model	99.85%	70.1%	67.6%
CNN Model with Increased Training	97.42%	70.9%	85%
Optimized CNN Model	98.97%	75.3%	89.1%
Pretrained Model 1	95.41%	91.8%	85.5%
Pretrained Model 2	94.76%	63.6%	57.2%
Pretrained Model 3	95%	97.5%	98.6%

Conclusion:

To sum up, this assignment investigated the effects of network design selection and training sample size on a two-class picture classification job. Overfitting was noted even though training CNN models from scratch yielded extremely high training accuracy. Network optimization and a larger training sample size improved test and validation accuracy. The efficiency of transfer learning in image classification was demonstrated by the noticeably better accuracies obtained when pretrained models (ResNet50V2, MobileNetV2, and VGG16) were used.