Assignment -2
Convolution Networks
CATS VS DOGS

Srinath Chowdary Alla **811250954**

ABSTRACT:

Convolutional Neural Networks (convnets) excel in image classification tasks, as seen in the Cats & Dogs example. This abstract explores two key approaches: training convnets from scratch and leveraging pretrained models. The challenge of overfitting with small datasets is addressed through techniques like data augmentation and regularization.

DATASET:

The Dogs vs. Cats Kaggle dataset is organized into two primary folders: 'train' and 'test.' The 'train' folder houses 25,000 labeled images of dogs and cats. These images and their corresponding labels (e.g., "dog" or "cat") are used to train your image classification model. The 'test' folder contains 12,500 images without labels. Your model's task is to predict the correct label (dog or cat) for each image in the 'test' set. The images are typically in JPEG format.

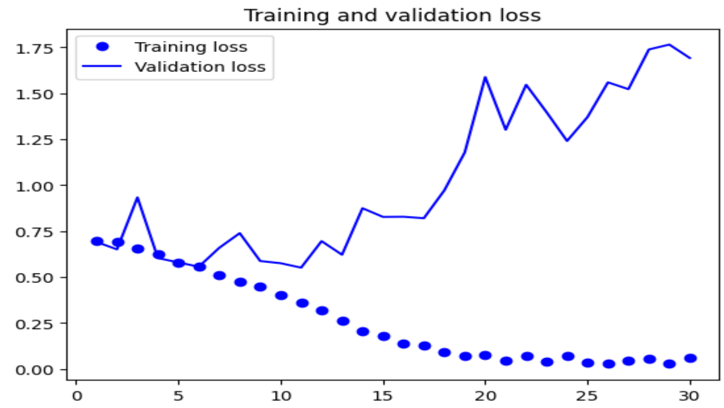
FINDINGS:

The models have been run with the following parameters to compile the model: optimizer='rmsprop', loss='binary_crossentropy,' metrics='accuracy'. At every step of the model augmentation and drop out layers were used to see if there are any increase the accuracy of the model.

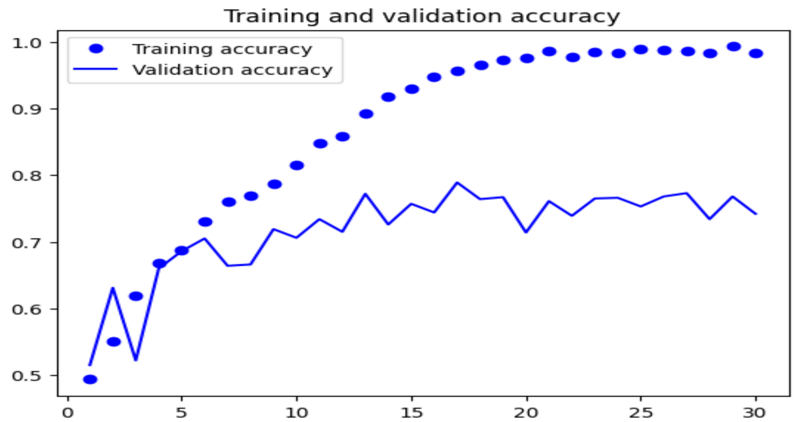
Question-1:

The base models in training samples of 1000, Validation samples of 500, and testing samples of 500.

The below image shows loss curves. The training loss starts high but decreases rapidly in the initial epochs, eventually converging to a very low value close to 0 by the end of training. The validation loss also decreases in the early epochs but then starts increasing again after around epoch 17, indicating the model is overfitting to the training data.



The below image accuracy curves. The training accuracy gradually improves from a low initial value, steadily increasing until it reaches very high values above 0.9 by the final epochs. The validation accuracy follows a similar increasing trend in the early stages but fluctuates between 0.7-0.8 after peaking around epoch 17. This divergence from the training accuracy again signals **overfitting.**



while the model can achieve very low training loss and high training accuracy by fitting the training data extremely well, the validation loss and accuracy curves reveal that after a certain point, it fails to generalize optimally to the unseen validation data. This is a classic case of overfitting.
The initial model with training samples of 1700, Validation samples of 500, and testing samples of 500 that yielded a test accuracy of **71%**.

```
63/63 [==============================] - 1s 14ms/step - loss: 0.5985 - accuracy: 0.7100
Test accuracy: 0.710
```

When model augmentation and drop out layers were used to the base model the test accuracy increased to **83.7%.**

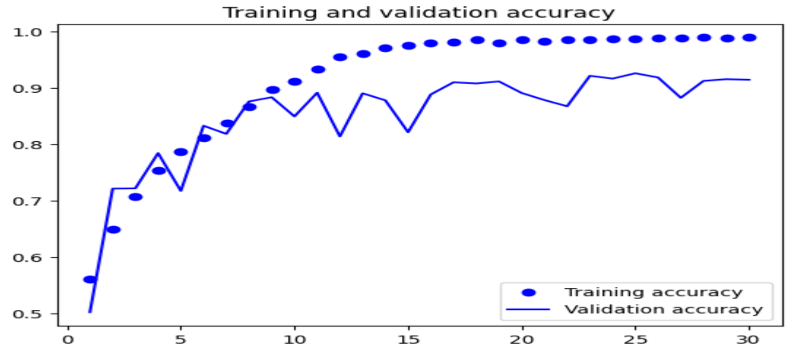
```
63/63 [==============================] - 1s 14ms/step - loss: 0.4445 - accuracy: 0.8375
Test accuracy: 0.837
```

Question-2:

The model of training samples has been increased to 3000, keeping the Validation samples of 1000, and testing samples of 1000 as above.

The training loss steadily decreases and converges to a very low value close to 0 by the end of training. However, the validation loss exhibits more fluctuations, with some increases indicating potential overfitting to the training data.

The training accuracy increases rapidly, reaching very high values above 0.95 by the final epochs, indicating the model has fit the training data very well. The validation accuracy also improves initially but fluctuates between 0.85-0.95 in the later epochs, not matching the high training accuracy consistently.



While the model achieves excellent performance on the training data by the end, as seen by the low training loss and high training accuracy, the deviations between the training and validation curves, especially in the later epochs, suggest some level of overfitting. This means the model may not generalize as well to unseen data, despite fitting the training set very closely. The training samples has been increased to 3000, keeping the Validation samples of 1000, and testing samples of 1000 that yielded a test accuracy of **86%**. As we can observe below.

```
125/125 [==============================] - 2s 14ms/step - loss: 0.3405 - accuracy: 0.8660
Test accuracy: 0.866
```

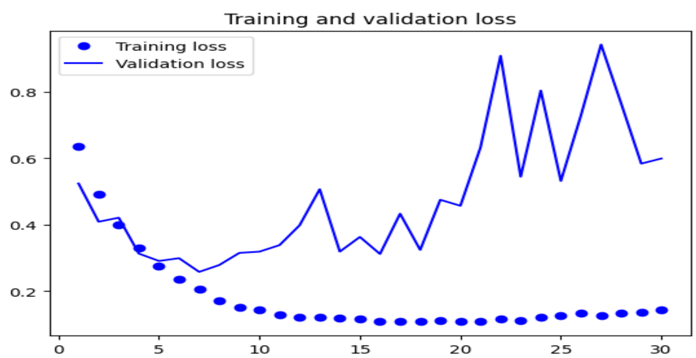
When model augmentation and drop out layers were used to the base model the test accuracy increased to **91.2%.**

```
125/125 [==============================] - 2s 14ms/step - loss: 0.2275 - accuracy: 0.9120
Test accuracy: 0.912
```

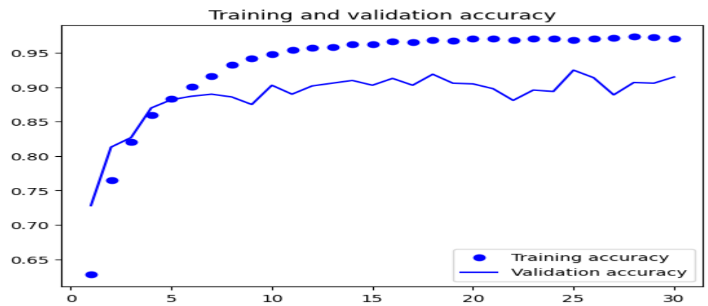
<u>Question-3:</u>

The model of training samples has been increased to 5000, keeping the Validation samples of 1000, and testing samples of 5000.Additionally using activation function **Sigmoda.**

In the loss plot, we see the training loss steadily decreasing, which is expected as the model learns from the training data over time. The validation loss exhibits more fluctuation, increasing sharply at certain points, which could indicate overfitting to the training data.



The accuracy plot shows a complementary trend - the training accuracy increases steadily as the model fits the training data better. The validation accuracy also increases overall but has more variation, dipping at certain points where the validation loss spikes. This suggests the model may be overfitting at times.



The model of training samples has been increased to 5000, keeping the Validation samples of 500, and testing samples of 5000 that yielded a test accuracy of **88%.**

```
32/32 [==============================] - 1s 14ms/step - loss: 0.3102 - accuracy: 0.8800
Test accuracy: 0.880
```

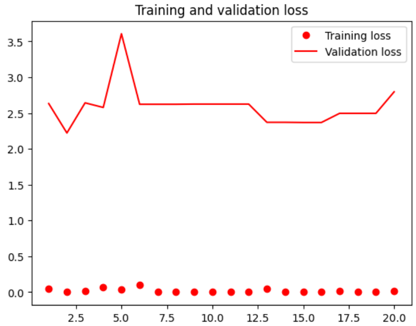
When model augmentation and drop out layers were used to the base model the test accuracy increased to **90.7%.**

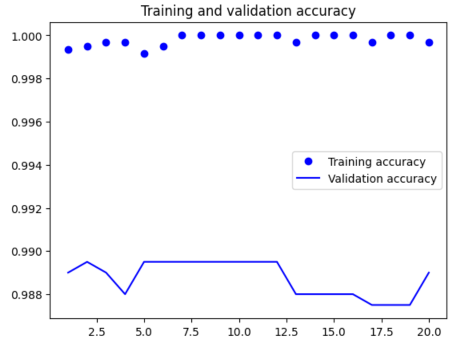
```
32/32 [==============================] - 1s 14ms/step - loss: 0.7373 - accuracy: 0.9070
Test accuracy: 0.907
```

<u>Question-4:</u>

The base models in varying training samples of sizes 2500,3000,4500,6000.In addition, we also use a pretrained network with feature extraction.

In the loss plot, we see that the training loss decreases sharply early on but then flattens out, while the validation loss remains relatively high and even increases towards the end. This suggests that the model may be overfitting to the training data, failing to generalize well to the validation set.
The accuracy plot tells a similar story. The training accuracy quickly reaches very high values close to 1.0, indicating that the model is essentially memorizing the training data. However, the validation accuracy remains significantly lower, again pointing to overfitting issues.
Interestingly, there are also periods where the validation loss spikes and accuracy drops, such as around iteration 3-4 and 16-17. This could indicate underfitting, where the model is struggling to capture the complexity of the data and failing to learn meaningful patterns.





The base models in varying training samples of sizes 2500,3000,4500,6000 yielded a test accuracy of **98.7%, 99.0%, 98.8% and 98.8%** respectively.

Conclusion:

In conclusion, this exploration of training convolutional neural networks (convnets) for image classification on the Dogs vs. Cats dataset has highlighted several key lessons. The initial base models with smaller training sets (1000-3000 samples) exhibited clear signs of overfitting, with validation metrics deviating significantly from training metrics after a certain point. Employing techniques like data augmentation and dropout helped mitigate overfitting to some extent and improved generalization performance.

Increasing the training set size to 5000 samples led to better results, with augmentation and dropout boosting the test accuracy to 90.7%. However, the most promising approach involved varying the training set sizes from 2500 to 6000 samples and leveraging a pretrained network with feature extraction. This strategy yielded remarkably high-test accuracies ranging from 98.7% to 99.0%, demonstrating the power of transfer learning and larger datasets in combating overfitting.

These findings underscore the importance of careful regularization, data augmentation, and sufficient training data when training convnets from scratch on relatively small datasets. Furthermore, the impressive performance achieved by leveraging pretrained models and larger training sets highlights the value of transfer learning and the availability of high-quality labeled data in achieving state-of-the-art results in computer vision tasks like image classification.

Conclusion

Your experiments reveal a recurring pattern of overfitting across various training dataset sizes and model configurations. Here's a breakdown of the key findings:

- **Overfitting Prevails:** In all scenarios, the models display clear signs of overfitting. This is indicated by the increasing divergence between training and validation loss curves, as well as the higher training accuracy compared to fluctuating or stagnating validation accuracy.
- **Data Quantity Matters:** Increasing training data slightly improves model performance (test accuracies of 86%-91.2%), but overfitting remains a significant challenge.
- **Model Complexity:** Augmentation and dropout layers provide some improvement but do not fully address the overfitting issue.
- **Pretrained Models with Feature Extraction:** Your experiment using a pretrained model with feature extraction achieved the highest test accuracies (up to 99.0%). This suggests that leveraging the knowledge embedded within pretrained models can be highly effective.

Recommendations

To combat overfitting and improve generalization, consider the following strategies:

- **Regularization:** Experiment with stronger regularization techniques like L1/L2 regularization and more aggressive dropout rates.
- **Data Augmentation:** Explore a wider range of augmentation strategies (rotations, flipping, color shifts, etc.) to increase dataset diversity and make the model more robust.
- **Early Stopping:** Implement early stopping to automatically terminate training when the validation loss starts consistently increasing.
- **Fine-tuning Pretrained Models:** Further explore pretrained models, fine-tuning not only the final layers but also deeper layers within the network.