Assignment - 2

Convolution Neural Networks

Jhansi Bussa

Hilda Gandu

Introduction:

Convolution neural networks are typically employed for input of images and feature extraction. This assignment investigates the effects of network selection and training sample size on model performance. To compare the effectiveness of training a network from scratch versus utilizing one that has already been trained, we can import the Cats & Dogs dataset from the Kaggle competition.

Dataset Overview:

The Cats & Dogs dataset consists of labeled images of cats and dogs.

There are several important steps in this assignment:

1. Training a network from scratch: A network is first trained from scratch using the set sample sizes. Several techniques are applied to reduce overfitting to improve performance.
2. Expanding the training sample size: Performance effects of expanding the training sample size are examined. The validation and test samples do not vary.
3. Optimizing network performance: The objective is to improve upon the previous steps by modifying the training sample size. This may need changing the sample size in order to strike the ideal balance.
4. Applying a pretrained network: The last phase is applying a pretrained network and carrying out Steps 2 and 3 once more. The usage of pretrained networks versus starting from scratch for performance improvement is examined in this step.

A relationship between the size of the training sample, the type of network, and overall performance is examined by recording and comparing outcomes, such as accuracy, at different steps of the proces.

Training , Test,Validation Accuracy based on Output results:

The CNN was trained on a substantial dataset containing numerous images of both classes. The model showed good accuracy on the training set as it learned to differentiate between the characteristics of cats and dogs. But the model performed worse when tested on a different dataset with photographs that had never been seen before; it only achieved an accuracy of about 69%. This shows that although the CNN was successful in learning from the training set, it had trouble applying what it had learnt to new images. The model may require additional optimization and tuning to improve its accuracy in classifying unseen data.

- Based on the provided output for train sample 1000, validation 500, Test 500:

Test accuracy: 68.95%

Validation accuracy: 75.10%

Training accuracy: 99.75%

These are the final accuracies obtained after training the model for 50 epochs.

The training accuracy of the model improved over time, starting at 52.25% in the first epoch and reaching 94.28% in the 32nd epoch. The validation accuracy also showed improvement, starting at 53.50% and reaching 86.40% in the 32nd epoch. The final test accuracy of the model was 83.80%.

- Based on the provided output for train 2000, test 500, validation 500:

Testing accuracy: 83.80%

Validation accuracy: 86.40%

Training accuracy: 94.28%

The training accuracy of the model improved over time, starting at 61.53% in the first epoch and reaching 96.14% in the 23rd epoch. The validation accuracy also showed improvement, starting at 70.50% and reaching 91.60% in the 23rd epoch. The final test accuracy of the model was 89.60%.

Training accuracy: 96.14%

Validation accuracy: 91.60%

Testing accuracy: 89.60%

The training accuracy of the model improved over time, starting at 96.02% in the first epoch and reaching 96.98% in the 7th epoch. The validation accuracy also showed improvement, starting at 91.40% and reaching 91.30% in the 7th epoch. The final test accuracy of the model was 89.70%.

- Based on the provided output for train 3000, test 500, validation 500:

Training accuracy: 96.98%

Validation accuracy: 91.30%

Testing accuracy: 89.70%

The model achievedd a high training accuracy of 96.30%, indicating that it learned the training data well. However, this high training accuracy did not translate well to the validation set, where the accuracy was only 61.00%.

- Based on the provided output for train 4000, test 500, validation 500:

Training accuracy: 95.79%

Validation accuracy: 98.20%

Test accuracy: 98.00%

The model's training accuracy tells us how well it learned from the examples it trained on, scoring about 96% correct. The validation accuracy, which checks how well it can handle new, similar example, was even higher at about 98%, showing it can generalize well. When tested on a fresh set of completely new examples, its accuracy remained high at about 98%.

Conclusion

In conclusion, the Cats & Dogs dataset provided valuable insights into the functioning of convolutional neural networks (CNNs) during our study. Although we were able to reach a final training accuracy of 96.30% when training CNNs from scratch, the validation accuracy remained lower at 61.00%, suggesting that there might be problems with overfitting.

The training and validation accuracies improved significantly with an increase in the training sample size, reaching 95.79% and 98.20%, respectively, with a matching test accuracy of 98.00%. By utilizing optimization strategies, the model's performance was further improved, resulting in a test accuracy of 89.70%, validation accuracy of 91.30%, and final training accuracy of 96.98%. Last but not least, using pretrained models produced the best training accuracy of 97.75%, but with somewhat lower test (68.90%) and validation (75.10%) accuracies, indicating significant limits

In generalization These findings emphasize the significance of pretrained models, optimization techniques, and dataset size in obtaining high-performance CNNs. However, they also point to the necessity of ongoing improvement to guarantee reliable generalization to new data.

Overall, there is a complex relationship that depends on a number of variables between the size of the training sample and network selection. Experimentation with different sample sizes, architectures, and optimization techniques is necessary to determine the ideal configuration that produces the best performance for a given task.