Assignment 2

Sai Sree Pulimamidi

Applying Convolutional Networks (Convnets) to Image Data

Introduction:

In this example we study the performance of developing a convolutional neural network using the cats and dog's dataset from Kaggle, determining which sample size and approach is most suitable throughout the model building stage.

Methodology:

In this we created 3 scratch models and two pre trained models.

Scratch models:

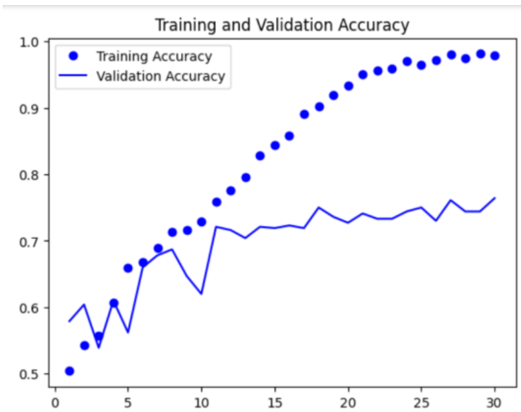
Accuracy, Validation Accuracy, Test Loss, Test Accuracy

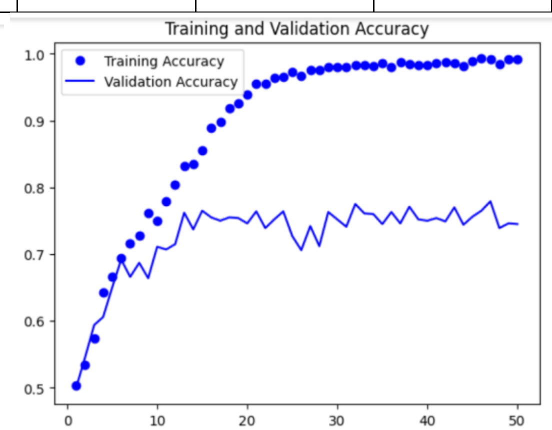
Model No	Training Sample Size	Validation, Test	Accuracy	Validation Accuracy	Test Accuracy	Test Loss
1	1000	500,500	97.85	76.40	69.20	62.83
2	1500	500,500	99.15	74.50	72.50	60.83
3	2000	500,500	99	72.80	68.80	63.04

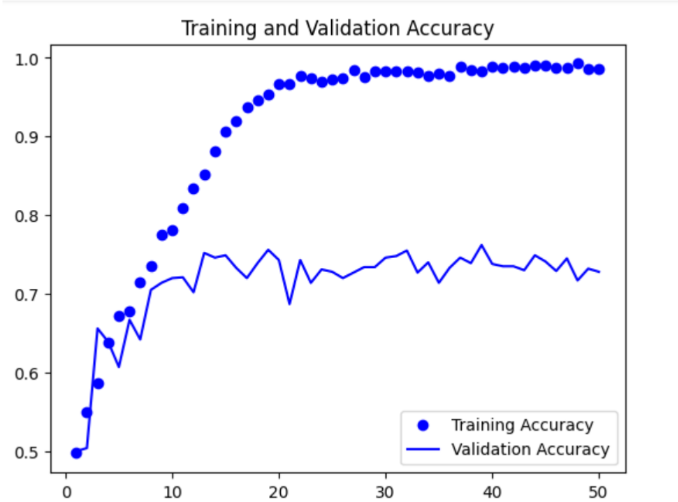
Pre-Trained Models:

Accuracy, Validation Accuracy, Test Loss, Test Accuracy

Model No	Training Sample Size	Validation, Test	Accuracy	Validation Accuracy	Test Accuracy	Test Loss
VGG Model -1	1000	500,500	97.20	97.60	96.10	0.49
VGG Model-2	1500	500,500	98.45	99	97.68	0.20







These are the results of three models constructed from the ground up, demonstrating that the larger the training sample size, the more accurate the model.

The best way to avoid overfitting is to change the training sample and use optimization techniques. It is not always possible to expand the training sample, so Data augmentation is a way to maximize training data.

When the model is smaller, i.e., in terms of how many learnable parameters there are in the model (i.e., how many layers and how many units in layers there are), the amount of overfitting is significantly reduced. By limiting the weights to very small values, you can reduce or prevent overfitting by normalizing the way the weight values are distributed. This makes the network less complex.

For optimization purposes, I have integrated learning rate and dropout approaches. The accuracy of the new model was higher than that of the prior model due to its bigger sample size and use of optimizers.

Conclusion:

In brief, the size of the training sample plays a crucial role in improving model accuracy by preventing overfitting. To further improve the model's performance, hyper-tuning elements like dropout strategy, and data augmentation are used.

1) A rather low accuracy of 70.90 is found in the Model 1 unregularized Model of Cats and Dogs example, which has 1000 training samples, 500 validation samples, and 500 test samples. This demonstrates overfitting because of the small training sample.
2) While maintaining the sample size at 500, we may be able to enhance the model's performance by mixing multiple methodologies. I used three different approaches on the model to do this: a) Drop out Technique.
   b) Drop-Out technique and data augmentation.
3) It was found that the model that was trained using the dropout technique and data augmentation performed better.
4) Train with additional data: Training with more data improves accuracy. We tried increasing the training samples to 1500 and 2000, and the accuracy increased.
5) When the models were pretrained, the accuracy increased to about **98%.**

Adding more data always helps increase data training, which boosts accuracy. As a result, we can see how a pre-trained network can be helpful in creating a better model with less data and greater accuracy due to the extensive training it received in the past.

Limitations:

1. The findings and ideal sample size may not be directly transferable to different datasets or problem domains.

2. While accuracy was used as the primary evaluation metric, other metrics like precision, recall, and F1 score could be more relevant for certain applications.

3. The computational resources available can significantly impact the choice of sample size and model complexity.

Recommendations:

1. Carefully analyze the dataset and problem to determine the appropriate sample size.

2. Use data augmentation and regularization techniques to reduce overfitting.

3. Experiment with various network architectures and hyperparameters to optimize performance.

4. Consider using pretrained networks when data is limited but be prepared to fine-tune them.

5. Continuously monitor model performance and iterate on the above steps to improve results.