Convolution Networks

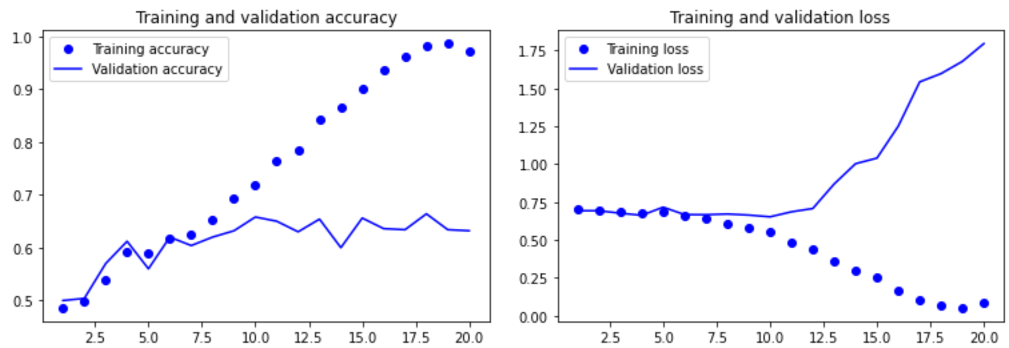
S.no	**Training Set**	**Validation Set**	**Test Set**	**Data augmentation**	**Pretrained Model**	**Loss and Accuracy on Test**
1.	1000	500	500	None	None	loss: 0.6202 - accuracy: 0.6420
2.	3000	500	500	None	None	loss: 0.5253 - accuracy: 0.7300
3.	16000	500	500	None	None	loss: 0.4193 - accuracy: 0.9020
4.	1000	500	500	Yes	None	loss: 0.6442 - accuracy: 0.6680
5.	3000	500	500	Yes	None	loss: 0.4832 - accuracy: 0.7600
6.	16000	500	500	Yes	None	loss: 0.2034 - accuracy: 0.9040
7.	1000	500	500	Yes	Yes	loss: 1.9545 - accuracy: 0.9820
8.	3000	500	500	Yes	Yes	loss: 1.9789 - accuracy: 0.9820
9.	16000	500	500	Yes	Yes	loss: 0.0871 - accuracy: 0.9840

Convolution Networks

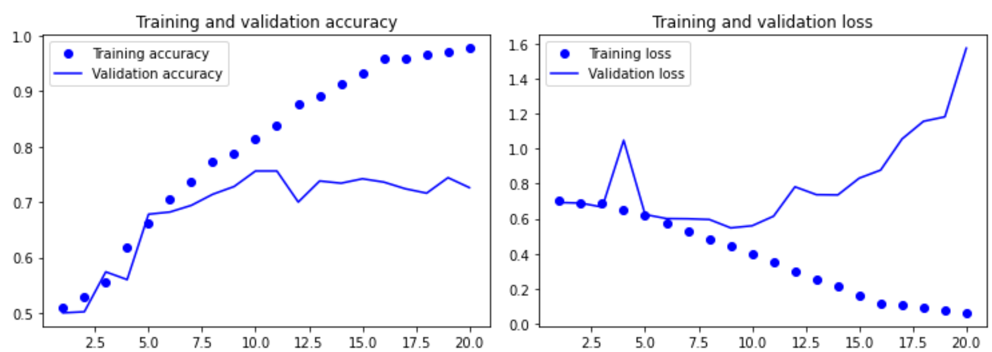
FINDINGS:

- Every model mentioned above has been used.• To construct the model, all of the aforementioned models were ran with the following parameters: metrics=['accuracy'], optimizer='adam,' loss='binary_crossentropy,').
- There are no extra parameters in the model to determine the ideal training sample size for the available dataset. The basic models demonstrated increasing accuracies with increasing training sample numbers in training samples of 1000, 3000, and 16000.
- Despite setting the validation and test samples to 500 for each iteration, we saw an increase in accuracy. This can occur as a result of the model picking up additional features from the training set of photos.
- With 1000 training samples, the accuracy was 0.6420 and the loss was 0.6202. the training accuracy grew over the course of 20 epochs, but the validation accuracy peaked at epoch four, at 0.61. the model was compiled using the following parameters: optimizer='adam, loss='binary_crossentropy,' metrics=['accuracy']).
- There are no extra parameters in the model to determine the ideal training sample size for the available dataset. The basic models demonstrated increasing accuracies with increasing training sample numbers in training samples of 1000, 3000, and 16000.
- Despite setting the validation and test samples to 500 for each iteration, we saw an increase in accuracy. This can occur as a result of the model picking up additional features from the training set of photos.
- With a base training sample size of 1000, the accuracy was 0.6420 and the loss was 0.6202. The findings below demonstrate that over the course of 20 epochs, the training accuracy rose with each epoch, but the validation accuracy peaked at 0.61% on the fourth epoch.

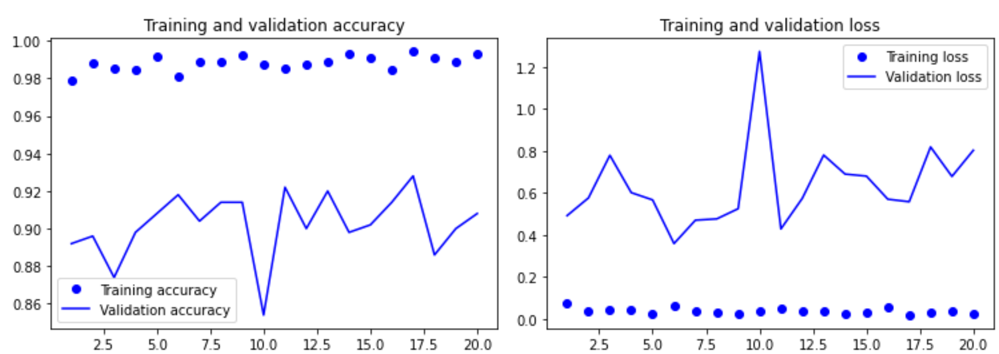
Convolution Networks



• The accuracy was 0.7300 and the loss was 0.5253 for the training samples of 3000. The test accuracy rose by 13% as the sample size climbed from 1000 to 3000. The findings below demonstrate that the validation accuracy peaked at 0.7560 at the tenth epoch.



• In training samples of 16000, the highest accuracy was attained at "0.9020". The accuracy of the model varied from 0.85 to 0.89 for each other training sample, that is, samples that were above and below 16000. This demonstrates that, despite the validation and test sets having constant sample sizes of 500 each, the model extracted the maximum features with training samples of 16000 and no further parameters used.



Convolution Networks

- The dataset can be processed using a variety of techniques to lessen overfitting. For the current dataset, there are two methods to improve the model's performance and lessen overfitting. They are pre-trained convents and data augmentation.
- Data augmentation uses a variety of random changes to produce additional samples from the training data that already exist. By preventing the model from ever seeing the same image twice, this aids in the learning of additional data features.
- A large dataset, often an extensive image classification dataset, is used to train a pre-trained network. The network's learned information can be used for different picture classification tasks, even if they are not part of the original dataset, if the dataset is large and diverse. A trained network can be used as a basis for several picture classification tasks.
- The following data augmentation settings have been applied to all models for the present dataset: levels.RandomFlip ("vertical"), strata.Rotate at Random(0.1),Random Zoom (0.2) layers.
- Data augmentation was used for each of the three basic models to raise the model's baseline performance. The following are the outcomes:
- O At the seventh epoch, the validation accuracy of the 1000 training samples peaked at 0.5140, and the test accuracy grew by just 4%, or from 0.6420 to 0.6680.
- o The test accuracy grew by just 4%, or from 0.7300 to 0.7600, while the validation accuracy of the 3000 training samples peaked at 0.6780 during the seventh epoch.
- o The test accuracy grew by just 0.2%, or from 0.9020 to 0.9040, after the validation accuracy of the 16000 training samples peaked at 0.8640 in the eighth epoch.
- Based on the results above, data augmentation performed worse for the current dataset than the base model, which demonstrated no appreciable gain in performance.
- We apply pre-trained convents with data augmentation and fine-tune the model for optimal performance in order to further increase the model's performance. For the current dataset, which consists of 1.4 million labelled images and 1000 distinct classes, we use the imageNet dataset, which was trained on the VGG16 architecture.

Convolution Networks

- To enhance performance in a pre-trained network, we employ two methods: 1. Extraction of features 2. Adjusting. The process of extracting important features from fresh samples by applying representations that a prior network has learned is known as feature extraction. After then, a fresh classifier that was built from scratch is applied to these features. Fine-tuning consists of unfreezing a few top layers of a frozen model base. for feature extraction and jointly training both the newly added part of the model (in this case, the fully connected classifier) and these top layers.
- By utilising feature extraction in conjunction with data augmentation and fine-tuning for each of the three models, we were able to attain the following outcomes:
  - At the fifth epoch, the validation accuracy of the 1000 training samples peaked at 0.9740, while the test accuracy rose by 47%, or from 0.6680 to 0.9820.
  - At the fifth epoch, the validation accuracy of the 3000 training samples peaked at 0.9800, and the test accuracy grew by just 29%, or from 0.7600 to 0.9820.
  - The test accuracy grew by just 8%, or from 0.9040 to 0.9840, whereas the validation accuracy of the 16000 training samples peaked at 0.9820 at the fifth epoch.

Conclusion:

The size of the training sample and the network selection for image classification are directly correlated. For image classification applications, deeper networks are needed to get improved performance as the training sample size grows. Larger training sample sizes further mitigate the problem of overfitting, enabling the construction of more complicated models without compromising their ability to generalise on unobserved data. Simpler networks can occasionally outperform more complex ones on image classification tasks, even with smaller training sample sizes. This may happen if there are few visual cues or patterns in the categorised photos that are simple for shallower networks to identify, or if the network design has been precisely tailored and optimised for the particular image classification issue.

We can see that when the number of training samples rose for the current dataset—which is image classification of dogs versus cats—the model performed better. 16000 is the ideal training sample, and the model performs at 0.9840. Apart from the size of the training sample and network architecture, performance in image classification is determined by multiple additional factors. This includes the optimisation process, the regularisation techniques used in training, and the quality of the pretreatment of the data. To guarantee that an image classification system is well-tuned and capable of achieving high accuracy on a variety of datasets, all these factors must be taken into account when constructing the system. Additionally, we included data augmentation and pre-trained convents in this model, which greatly enhanced. Thus, when creating an image classification system, it is essential to take into account each of these elements in order to produce reliable and accurate results.