<u>Assignment-2 Convolution</u>

Varshitha Guduru

Report on Pretrained Convolution Networks vs. Scratch Model Training

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 (like in the text). Use any technique**
   to reduce overfitting and improve performance in developing a network that you train
   from scratch. What performance did you achieve?

A small dataset consisting of 1000 training samples, 500 validation samples, and 500 test samples was used to train the first model. The outcomes were as follows:
Test accuracy = 70.1%.
Methods like data augmentation and dropout layers were used to improve the model and lessen overfitting. Although the model's performance was improved, these techniques still highlighted the drawbacks of starting from scratch and using a small dataset for training.

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 validation and test sets remained unchanged, however the training sample was increased from 1000 to 1500. As a result, the model was able to learn from more samples and achieve the following improvements:
Test accuracy: 68.5%.
Nevertheless, the model continued to exhibit overfitting, and the test accuracy was not considerably increased by this increase in the training sample size.

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.

Following trial and error with various sample sizes, the training set was expanded to 1500, accompanied by a test set of 500 and a validation set of 1000. Following optimization, the outcome was:
Test accuracy:70.5%.

To boost generalization and artificially expand the training data, data augmentation techniques were used. However, the minor gains in performance suggest that more improvements may require more advanced approaches or a different approach**.**

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.**

In order to compare performance with the scratch-trained model, the pretrained VGG16 network was presented. To extract features, the VGG16 model was loaded onto ImageNet with pretrained weights. With identical test, validation, and training set sizes, the outcomes were:

Test accuracy: 67.4%.
Compared to training from scratch with the same sample size, this result was somewhat worse. On previously unreported data, the VGG16 model did, however, enable faster convergence and improved generalization.

TRAINING SETUP	TEST ACCURACY	STRENGTHS
From Scratch (Initial)	70.1%	Excellent performance, although with little amounts of data, prone to overfitting.
From Scratch (Increased Data)	68.5%	A slight decrease in performance, demonstrating that additional data by itself is insufficient without appropriate optimization.
Optimized (Scratch)	70.5%	Maximum performance, but still at risk of overfitting and requiring careful adjustment.
Pretrained(VGG16)	67.4%	On limited datasets, lower accuracy but greater stability and resistance to overfitting.

CONCLUSION:

Finally, because pretrained networks use learned features from big, diverse datasets like ImageNet, they are very effective for smaller datasets and enable faster training and improved generalization. They require less computing power and fine-tuning to get respectable performance, and they are less prone to overfitting. This makes them especially appropriate in situations when there is a shortage of data or when prompt deployment is required.

However, when larger datasets are available, training from scratch can perform better than pretrained models. While training models from scratch, suitable regularization approaches like dropout and data augmentation along with adequate hyperparameter optimization can lead to somewhat improved accuracy. To guarantee convergence and avoid overfitting, this method necessitates careful optimization, greater processing power, and more time. Essentially, the size of the dataset, the available processing power, and the intended performance should be taken into consideration while deciding between pretrained and scratch training.

Key Findings

1. Training from Scratch can produce somewhat better accuracy, however in order to avoid overfitting, more tweaking and larger datasets are needed.
2. Pretrained networks are useful in situations where a quick convergence is required or the dataset is tiny, even though they might provide significantly less accuracy.
3. Optimal training sample size based on the amount of the dataset and the complexity of the model. More data is beneficial, but only in conjunction with effective optimization techniques.

In conclusion, while scratch-trained models can perform better with larger datasets and meticulous optimization, pretrained models are a good option for quicker, more stable results, especially with little data.