**ASSIGNMENT-2**

**Exploring the Impact of Training Sample Sizes on Convolutional Neural Networks**

**PROBLEM STATEMENT**

In this assignment, we aim to explore the relationship between training sample size and the choice of training a convolutional neural network (CNN) from scratch versus using a pretrained network. Specifically, using the Cats & Dogs example, we will investigate the performance of these two approaches across varying training sample sizes. By analyzing the impact of sample size on model performance, we seek to understand the optimal conditions for achieving the best prediction results.

**INTRODUCTION**

In this report, the relationship between training sample size and the choice of network architecture (training from scratch vs. pretrained networks) is investigated using the Cats & Dogs image classification example. The objective is to understand how varying training sample sizes affect model performance and to determine the optimal approach for achieving accurate predictions.

**METHODOLOGY**

1. **Training from Scratch**

* Initial training sample size: 1000 images
* Validation sample size: 500 images
* Test sample size: 500 images
* Techniques used:

**Data augmentation:** Increasing the variety of training data by changing aspects like rotation, flipping, and scaling.

**Regularization:** Adding extra terms to the model's loss function to prevent it from becoming too complex and fitting too closely to the training data.

1. **Increasing Training Sample Size**

* Training sample size increased beyond 1000 images while keeping validation and test samples constant.
* Application of optimization techniques to fine-tune the model.

1. **Optimal Training Sample Size**

* Experimentation with different training sample sizes to identify the ideal size for optimal performance.

1. **Pretrained Networks**

* Repeat Steps 1-3 using a pretrained convnet, exploring different training sample sizes.
* Flexibility in adjusting sample sizes to observe performance differences.

**TECHNIQUES**

**Preprocessing**

1. Access the image files.
2. Decode the JPEG content to obtain RGB pixel grids.
3. Convert the pixel grids into floating-point tensors.
4. Normalize the pixel values to fit within the [0, 1] range, as neural networks perform better with smaller input values. This is achieved by scaling the pixel values, which originally range from 0 to 255.

**Data Augmentation**

We aim to utilize data augmentation techniques to enhance the accuracy of our model. Data augmentation involves generating new data from existing training samples by introducing random variations, allowing for improved performance even with limited datasets. By applying this method, the model is exposed to different versions of images during training that it hasn't encountered before, facilitating better generalization. To achieve our objective, we intend to randomly apply various transformations such as flipping, rotating, and zooming to the images in the training set. This approach generates diverse versions of the original images, thereby enriching the dataset and strengthening the resilience of our model.

**Pre-trained model**

This dataset encompasses a wide array of animal classifications, including various breeds of dogs and cats. A prominent example of a simple convolutional neural network architecture suitable for such tasks is VGG16, which has been widely used in image recognition tasks like ImageNet. Leveraging a pretrained network like VGG16 on a large and diverse original dataset allows for the extraction of generic features applicable to a range of computer vision applications. Deep learning's ability to transfer learned characteristics across different tasks stands as a significant advantage over traditional machine learning methods. For instance, examining a large-scale trained convolutional neural network using the ImageNet dataset, which comprises 1,000 distinct classes and 1.4 million annotated images, showcases this capability.

In applying a pretrained network, two primary methods are commonly employed: feature extraction and fine-tuning. In this context, we'll focus on feature extraction to enhance outcomes. Initially, we'll extract features without data augmentation and subsequently incorporate augmented data to further refine the results. This approach capitalizes on the pretrained network's ability to extract meaningful features from the dataset, thereby improving the model's performance in subsequent tasks.

**RESULTS**

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?**

To tackle potential overfitting in the Cats & Dogs dataset, I used a training set of **1000** samples, with **500** for validation and **500** for testing. Given the risk of overfitting with this training size, I implemented a dropout strategy of **50%**.

**Hyper tuning Parameters:**

A graph with blue dots

Description automatically generated

A graph with blue lines

Description automatically generated

|  |  |
| --- | --- |
| **Validation accuracy(%)** | **Test accuracy(%)** |
| 74.60 | 74.20 |

1. **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?**

A collage of a cat

Description automatically generated

|  |  |
| --- | --- |
| **Validation accuracy(%)** | **Test accuracy(%)** |
| 81 | 79.6 |

The outcomes show significant improvement compared to the previous results (Question 1).

Our decision to increase the training sample size from **1000** to **1500** has notably enhanced the model's performance. Both training and validation accuracies have increased by over **5%**, indicating a substantial improvement. Additionally, besides employing convolution layers, we integrated data augmentation techniques. This inclusion contributed to refining feature extractions and ultimately led to superior performance.

1. **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.**

Increasing the amount of training data is commonly known to enhance model performance, but determining the optimal sample size can be challenging. In this case, incorporating data augmentation techniques and adding **500** samples to the dataset notably improved the model's performance, boosting it from **83%** to **80.1%.** However, despite the enriched data and larger sample size within the specified convolutional architecture, the model exhibits limited capacity to assimilate new information, highlighting a clear instance of this phenomenon. This discovery suggests the need to explore alternative approaches for enhancing the model's performance.

1. **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.**

**Pre-Trained model without Augmentation**

The model attained a validation accuracy of **97.9%** and a test accuracy of **97%**. While the test accuracy is promising compared to the initial training of a smaller model, there's a worrisome pattern of overfitting. Plots vividly depict this overfitting, despite implementing dropout regularization at a relatively high dropout rate.

Although the dropout plots imply early occurrence of overfitting in the training phase, indicating potential challenges in generalizing to unseen data, the model performs admirably on the validation data (used for fine-tuning hyperparameters).

A graph with blue dots

Description automatically generated

A graph with blue dots

Description automatically generated

**Pre-Trained model with Data Augmentation:**

Selecting the appropriate data for evaluating a model is crucial because the performance on one dataset may not necessarily generalize to others due to differences in complexity. This is exemplified by the accuracy of the pre-trained model, which stood at **97.9%** without data augmentation and **97.7%** with data augmentation. These slight variations underscore the importance of considering the specific characteristics and intricacies of each dataset. While a high accuracy rate may be achieved on one dataset, it doesn't guarantee similar success on others. Therefore, it's essential to exercise caution and thoroughly assess model performance across various datasets to ensure its robustness and generalizability.

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Training Samples** | **Validation Accuracy** | **Test Accuracy** |
| Model 1 | 1000 | 74.6 | 74.2 |
| Model 2 | 1500 | 81 | 79.6 |
| Model 3 | 2000 | 83 | 80.1 |
| Model 4 | Pretrained Model without data augmentation | 97.9 | 97.5 |
| Model 4 | Pretrained Model with data augmentation | 97.9 | 97.7 |

**DECISION**

- The choice of network (training from scratch vs. pretrained) impacts performance differently based on training sample size.

- Increasing training sample size generally leads to improved performance.

- Pretrained networks may offer better performance with smaller training sample sizes due to transfer learning.

**RECOMMENDATIONS**

- When limited by a small dataset, consider using pretrained networks to leverage transfer learning.

- Experiment with different training sample sizes to find the optimal balance between data availability and model performance.

- Implement techniques like data augmentation and regularization to mitigate overfitting.

**SUMMARY**

In this study, we investigated the relationship between training sample size and the choice of network architecture in image classification tasks. Through experiments with the Cats & Dogs dataset, we demonstrated the impact of sample size on model performance and highlighted the advantages and limitations of training from scratch versus using pretrained networks.

**CONCLUSION**

The relationship between training sample size and choice of network architecture is crucial in determining model performance. While larger training sample sizes generally lead to better performance, the effectiveness of pretrained networks with smaller sample sizes highlights the importance of transfer learning. Experimentation and optimization are essential to identify the most effective approach for a specific task and dataset.