DEEP LEARNING FOR COMPUTER VISION

Spring-2023 **AML Report** **Niharika Kolliboyana**

Executive Summary

The aim of our project is to build a new convolutional neural network from scratch, which is specifically designed for computer vision tasks. The dataset we are working with is a subset of the "Dog-vs-Cats" dataset available on Kaggle. We have limited data available, which makes the development of an effective model a challenging task.

Convolutional neural networks, or convnets, are a popular type of deep learning model that have proven to be highly effective in computer vision tasks. One of the key advantages of convnets is their ability to learn and detect spatial patterns within images. This makes them well-suited for tasks such as image recognition, object detection, and segmentation.

Despite the limited data available, we believe that our convnet model can still produce reasonable results. This is due to the ability of convnets to learn and generalize from small datasets by extracting and identifying relevant features from images. We plan to train our model on the limited dataset, fine-tune it using transfer learning techniques, and validate its performance using appropriate evaluation metrics. Overall, our goal is to develop an accurate and efficient convolutional neural network that can effectively classify images from the "Dog-vs-Cats" dataset with a limited amount of data.

Problem

The Cats-vs-Dogs dataset is a binary classification problem where the goal is to predict whether an image belongs to the dog class or a cat class.

Techniques

Dataset:

The Cats-vs-Dogs dataset contains 25,000 images of dogs and cats (12,500 from each class) and is 543MB large (compressed). After downloading and uncompressing it, we will create a new dataset containing three subsets: a training set with 1000 samples of each class, a validation set with 500 samples of each class, and finally a test set with 500 samples of each class. Due to the larger image size and more complex nature of the problem we are working on, we need to increase the size of our neural network. To accomplish this, we will add an extra stage to our existing Conv2D + MaxPooling2D architecture. This will provide additional network capacity, while also reducing the size of the feature maps, ensuring that they are not too large when we reach the Flatten layer. Our input images are initially 150x150 in size, and as we progress through the layers of the network, the feature maps gradually decrease in size until they are 7x7 right before the Flatten layer. This choice of input size is somewhat arbitrary, but is appropriate for the given problem.

Preprocessing:

- Read the picture files.
- Decode the JPEG content to RBG grids of pixels.
- Convert these into floating point tensors.
- Rescale the pixel values (between 0 and 255) to the [0, 1] interval (as you know, neural networks prefer to deal with small input values).

Data Augmentation:

To enhance the accuracy of our model, we intend to use data augmentation techniques. Data augmentation enables us to obtain satisfactory results even with limited datasets by generating additional data from the available training samples through random transformations. As a result, the model will never encounter the same image twice during training, which helps to improve its generalization capability.

For our specific task, we plan to randomly apply transformations such as flipping, rotating, and zooming to the images in the training data. By doing so, we can create variations of the existing images, thereby increasing the diversity of the dataset and improving the robustness of our model.

Pre-trained model:

If the original dataset is both extensive and varied, a pretrained network can be utilized as a generic model, and its features can be applied to many different computer vision tasks. This capability to transfer learned features across different tasks is one of the key strengths of deep learning when compared to other machine learning techniques.

As an example, we can consider a large convolutional neural network that has been trained on the ImageNet dataset, which contains 1.4 million labeled images and 1,000 different classes. This dataset includes several animal classes, such as various breeds of cats and dogs. The architecture of this network is called VGG16, which is a simple and widely used convnet architecture for ImageNet.

There are two primary methods for utilizing a pretrained network: feature extraction and fine-tuning. In this, we will use feature extraction, first without data augmentation, and then with data augmentation to achieve even better results.

Results: The table below shows the accuracy and validation loss for each approach.

TABLE FOR MODEL FROM SCRATCH

Train Size	Test Size	Validation Size	Data Augmentation	Train Accuracy(%)	Validation Accuracy(%)
1000	500	500	NO	77.4	71.2
1000	500	500	YES	73.2	73
1500	500	500	NO	84.5	73.6
1500	500	500	YES	73.1	70.9
1500	1000	500	YES	85.6	73.7
1500	1000	500	NO	53.7	54.5

TABLE FOR PRE-TRAINED MODEL

Data Augmentation	Train Accuracy(%)	Validation Accuracy(%)
NO	99.8	97.3
YES	96.9	97.4

In the tables above, the model configurations are listed, along with the sample sizes for the train, test, and validation sets. For the model from scratch, we include results with and without data augmentation and for models trained with an increase in train size or with different train and validation sizes. For the pre-trained model, we compare the accuracy, validation accuracy, with and without data augmentation.

Based on the results, we can observe that the models trained with data augmentation consistently were not able to perform better than those trained without it. Additionally, increasing the size of the training set or adjusting the size of the validation set also improves the accuracy of the model. Comparing the pre-trained model with and without data augmentation, we can see that data augmentation did not lead to an improvement in the accuracy, validation accuracy of the model. Overall, pre-trained models tend to outperform models trained from scratch, particularly when dealing with limited training data.