Machine Learning Engineer Nanodegree

Capstone Proposal

Edson Romero
May 18, 2020

Domain Background

In this project I will use convolutional neural networks to identify dog breeds. Recent research in the past 5 years have made break throughs in image classification by using convolutional layers. These layers can extract features from images by using a kernel and applying convolutional operations. These features can be high and low level where high-level features signify edges and shapes of the semantic object and low-level features signify finer details such as eyes, nose, and mouth of a dog for example. Generally, the more convolutional layers and the more kernels per convolution we use the more features we can extract from images. This leads to big network architectures, commonly refer to as deep networks, which results in significant increases in training time, even when high-end computational resources are available. To reduce training time researchers have developed a technique known as transfer learning where a deep network has been trained on a large dataset, and to reuse this training we replace some layers towards the end of the network to fit our custom dataset and retrain. The idea is that this new network will have a better starting point during training and hence will achieve better performance than a similar network trained for the same number of epochs. To solve the dog breed classification problem, I will use transfer learning by exploiting the architecture of the VGG16 model pretrained on the ILSVRC-2014 dataset which is a subset of ImageNet containing 1.2 million images and 1000 categories.

Problem Statement

Our problem of dog breed classification is very specialized as the model would have to learn to distinguish the difference between similar looking animals. That is, every dog has eyes, a nose, ears, fur, four legs and some breeds can be very difficult to tell apart for the human eye. Thus, for application purposes, we desire our model to achieve superhuman performance. By using deep networks and transfer learning I hope to achieve this goal. As stated one possible solution can be to use the VGG16 model pre-trained on the ILSVRC-2014 dataset. This model was trained for 2-3 weeks using four NVIDIA Titan Black GPUs and achieved first place in the ImageNet classification task challenge in 2014 with an accuracy of 92.7% [1]. Luckily, it is packaged up in PyTorch and hence is readily available for transfer learning for our dog breed dataset. The model will be evaluated by its accuracy score on the classification task and will be compared to a benchmark model as a sanity check. The benchmark model will be a network built from scratch using PyTorch with a similar architecture as VGG16 but much smaller in size. It will be trained for the same number of epochs and with the same optimizer as the pre-trained model. The expectation is that the pre-trained model should easily outperformed the benchmark model.

Datasets and Inputs

The breed dog dataset has 20,580 images and 113 categories where there are approximately 150 images per category [2]. The dataset was obtained as a subset of the repository of ImageNet, the repository contains roughly 14 million images. This dog dataset gives rise to a specialization challenge in deep learning where images are fairly similar to one another. For example, it would be difficult for an untrained human to recognize the difference between Norfolk Terriers and Norwich Terriers. The dataset will be subdivided into training, validation and testing set. The validation set will be used for tuning hyperparameters such as the learning rate. The test set will be used to evaluate the performance of the model.

Solution Statement

The solution will be to use transfer learning with the VGG16 model. The model was trained on a large dataset and the dog breed dataset is small. Both these datasets are somewhat similar as they contain four-legged animals. Therefore, the type of transfer learning we will use is that of replacing the last layer with a new layer of the same number of neurons as the number of dog breeds. Then all layer weights will remain fixed during training except those of the last layer. This is done to avoid overfitting to the dog dataset which is a problem with smaller datasets. Also, since the datasets are similar, high end and low-level features extracted from convolutions should be transferable from one model to another. In order to evaluate model performance, a benchmark model will be trained from scratch with the expectation that it should perform worse than the pre-trained model.

Benchmark Model

The benchmark model will be built using PyTorch and trained for the same number of epochs and with the same optimizer as the pre-trained model. We expect the pre-trained model to easily outperform the benchmark model in accuracy. If this is the case, we can then try to improve the benchmark model to see how far it can go. By using techniques such as image augmentation and dropout to improve validation accuracy, and batch normalization and modern optimizers such as Adam and RMSprop for faster learning we can test to see how close the benchmark model can get to the pre-trained model in accuracy. The architecture will be similar in design to that of VGG16 but smaller since we don’t have as powerful GPUs in the Udacity workspace and don’t want to train for 2-3 weeks. The architecture of VGG16 is of 5 convolutional sets each followed by a max-pooling layer which is then followed by 3 dense layers. Each convolutional layer increases the channel size of the image features going from 64 channels of dimensions 224 x 224 to 512 channels of dimensions 7 x 7 by the end of the convolutional layers. In a simpler design, the benchmark model will have 3 convolutional layers each followed by a max pool layer and increasing the channel size from 16 to 32 and then to 64. This is followed by 3 dense layers of sizes 1024, 512 and 113.

Evaluation Metrics

The evaluation metric of choice is accuracy for both the benchmark model and the pre-trained model. This metric is appropriate as it is used in evaluating models the classification challenge of ImageNet.

Project Design

Since the dataset is already obtained and annotated from a reputable source, ImageNet, the workflow of the project begins with the transformation of the dog breed dataset in order to feed it into the models. Aside from image resizing and normalizing, I plan to use image augmentation such as translation, rotation, shearing to see if performance can be improved. Then both the pre-trained and benchmark model will be trained for the same number of epochs and with the same optimizer so that they are comparable. The expectation will be that the pre-trained model will outperform the benchmark model because of transfer learning. If realized, I will then attempt to improve the benchmark model as much as possible to see how close it can get in accuracy to the pre-trained model. Note that the main architecture of the benchmark model won't change but supportive layers such as dropout and batch normalization will be added. This will be done so that we can obtain the best benchmark model given the resources such as GPUs and learning techniques. Then the difference in accuracy between the pre-trained and the improved benchmark model will demonstrate the importance that transfer learning has on deep learning.

Citations:

1. https://arxiv.org/pdf/1409.1556.pdf
2. http://vision.stanford.edu/aditya86/ImageNetDogs/