Jarryd Stringfellow

n9734074

Chase Dart

n

Abstract

The results that were successfully found in this report indicate that

Transfer learning assignment

CAB320 – Artificial Intelligence

Assignment 2

Semester 1, 2022

Table of Contents

[Introduction 1](#_Toc106284614)

[Research Process 2](#_Toc106284615)

[1. Preparing Dataset 2](#_Toc106284616)

[2. Selecting the Pretrained Model 2](#_Toc106284617)

[3. Implement a Dense Layer 2](#_Toc106284618)

[4. Prepare Training, Validation and Test Sets 2](#_Toc106284619)

[5. Compilation and Training Using Optimiser 3](#_Toc106284620)

[Results and Recommendations 3](#_Toc106284621)

[6. Training and Validation Errors and Accuracies 3](#_Toc106284622)

[7. Modifying the Orders of Magnitude 3](#_Toc106284623)

[8. Adding Momentum to the SGD Optimiser 4](#_Toc106284624)

# Introduction

The aim of this assignment is to build a flower classifier using transfer learning on a neural network trained on the ImageNet dataset. A range of flowers including a daisy, dandelion, rose, sunflower, and tulip have been chosen for this task. This task will be undertaken by coding in the Python language. Using the pre-trained dataset provided, the goal is to generate a process to be able to identify the flowers provided. The learning process will be documented in this report to show the steps that have been taken.

Transfer learning is a machine learning method where we reuse a model trained on a first dataset called the source dataset as the starting point for training a model on a second dataset called the target dataset. Generally, the source dataset is a large dataset like MobileNetV2 and the target dataset is a much smaller dataset relevant to a new application domain.

The method of applying transfer learning in through deep learning is as such:

1. A previously trained model will be sourced to remove the initial heavy lifting. This layers from this model will be used.
2. To avoid destroying any information they contain, the weights are frozen.
3. New trainable layers are added above the frozen layers. These are the layers that will learn to turn the old features into predictions on a new dataset.
4. Train the new layers on the provided dataset.

# Research Process

## Preparing Dataset

The dataset is comprised of five different flower types: daisy, rose, tulip, sunflower, and dandelions. Within each category, two-hundred unique images of the object have been selected. Each image has the specific flower as the primary medium of the image.

Each of the images are broken into their relative folder for ease of access during the process. For this assignment, the images are in a dataset folder one layer outside the python file. This was to allow the data to be accessible and not require the large quantity of files top be sent to GitHub for collaboration.

A white flower with a yellow center

Description automatically generated A bee on a yellow flower

Description automatically generated A picture containing plant, flower, rose, begonia

Description automatically generated A picture containing tree, outdoor, plant, flower

Description automatically generated A close up of a flower

Description automatically generated with low confidence

Figure 1: Example of each flower type

## Selecting the Pretrained Model

The pretrained model that was chosen was the MobileNet V2. This dataset was chosen as it is the smallest dataset within the Keras opensource library that still maintains a high Top-1 accuracy (A correct result).

The information about the model can be seen below:

| **Model** | **Size (MB)** | **Top-1 Accuracy** | **Top-5 Accuracy** | **Parameters** | **Depth** | **Time (ms) per inference step (CPU)** | **Time (ms) per inference step (GPU)** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| [MobileNetV2](https://keras.io/api/applications/mobilenet/#mobilenetv2-function) | 14 | 71.3% | 90.1% | 3.5M | 105 | 25.9 | 3.8 |

Table 1: Keras model data

## Implement a Dense Layer

Replace the last layer with a Dense layer of the appropriate shape given that there are 5 classes in the small flower dataset.

## Prepare Training, Validation and Test Sets

To use the data, the dataset must be broken up into their respective sets for the application. This is done by splitting up the images into a category.

The first dataset is training. This is where the images will be passed through the classifier to help train the new model.

The next category is the validation category. This provides the image with a known identifier for the model to check its accuracy by providing unbiased evaluation of a model fit for the training dataset. It also helps to tune the model hyperparameters.

Finally, the testing dataset provides an unbiased evaluation of a final model fit on the training dataset.

## Compilation and Training Using Optimiser

Compile and train your model with an SGD3 optimizer using the following parameters learning\_rate=0.01, momentum=0.0, nesterov=False.

# Results and Recommendations

## Training and Validation Errors and Accuracies

Plot the training and validation errors vs time as well as the training and validation accuracies.

Graphical user interface, application, Teams

Description automatically generated

## Modifying the Orders of Magnitude

Experiment with 3 different orders of magnitude for the learning rate. Plot the results, draw conclusions.

Chart, line chart

Description automatically generated

## Adding Momentum to the SGD Optimiser

With the best learning rate that you found in the previous task, add a non zero momentum to the training with the SGD optimizer (consider 3 values for the momentum). Report how your results change.

Chart, line chart

Description automatically generated