Machine Learning

Trainer

Summary: Introduction to convolutional neural networks and the Tensorflow Library

Contents

Chapter I
Foreword

The Human mind is incredible. It can look at two completely unrelated images with no two pixels being alike and determine in less than a second whether or not that blob of information is a tiger, the number 3, or a painting.

That was the basis for MADALINE. Bernard Widrow and his graduate student Ted Hoff believed that it was possible to create a simple version of a mind that could, through iterative tweaking, memorise patterns and predict what would follow a certain set of data.

Chapter II

Introduction

Writing functions is challenging. Especially when you start to deal with hundreds to thousands of parameters.

You have to consider how each parameter should affect the outcome. Which parameters are irrelevant and which are important. If they are important, how so? Should it be more or less prominent?

This is where machine learning comes in. It is much easier to write a function to check if a function performs the required task than it is to create the function in the first place. Machine learning uses this concept at its core.

Chapter III

Goals

For this project, Trainer, the aim is to understand how to build a neural model and what types of layers are most useful for what tasks. You will need to learn how to train a network and how to use that trained model.

Chapter IV

General Instructions

You will be given a database of clothing and you will need to create a neural network that, after some training, will be able to identify what types of clothing are in any given image with some accuracy.

Your program will be written in the latest version of Python3.

Allowed Python libraries are:
• Tensorflow

Your work will be tested with a test database of images during corrections. You may not use this database to train your images. You may only use it to test your trained model.

Chapter V

Mandatory Part

V.1 The Brains

Create a program that will make and compile a model that will be able to use the training data to learn. Save it to a file called 'base_model.h5' in the root of your repository.

Your model must have 10 output nodes for how likely it thinks the image is to belong in each category

Categories are as follows:

0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

V.2 The Gains

Write a program that will load your model and train it to recognise clothing using the provided database file then save it to a file called 'trained_model.h5' in the root of your repository.

V.3 All grown up

Write a program that, using the trained_model, will test the accuracy of the model against the test data set. it should print out each type of clothing and how often the model predicted correctly, followed by an overall accuracy percentage.

ex. "
T-shirt/Top: 80.0%
Trouser: 85.3%
.
.
.
Ankle boot: 75.4%

Overall: 82.5%
"

Chapter VI

Bonus Part

VI.1 Accuracy

Achieve (90%) accuracy for as many categories as possible

VI.2 Custom Data

Add your own data to the testing and training data sets.

Chapter VII

Turn-in and peer-evaluation

Turn in your work using your GiT repository, as usual. Only the work that's in your repository will be graded during the evaluation.

Each part of the mandatory part will be under a folder called the_brains, the_gains, and all_grown_up respectively and will start in a file called 'main.py' within each directory.

You must push your model files in root of the repository.

All scripts will be run from the root of your repository.

You must also push a diagram which shows the structure of your neural network.