Genetic Programming/Auto-ML for One-Shot Learning

Lorenzo Paris
University of Essex,
Department of Computer Science and Electronic Engineering,
Data Science and Decision Making, Assignment2, April 2018

***GitHub Repository**: https://github.com/Elleppi/ce888labs/tree/master/assignment*

***Abstract*—Are computers able to learn and classify from only one sample just like humans do? Will they able to categorise in ways that are mostly indistinguishable from people? A model example of such capabilities is the one-shot learning method, which correctly makes predictions by using only a single example of each new class. In this paper, I explore a learning method based on Genetic Programming which makes use of TPOT, a Python Automated Machine Learning tool. The One-Shot learning approach in this work will concern characters recognition. By having a small set of character samples, it will be able to recognise whether pairs of random characters written by different people are actually the same character, or not. Similar efforts have been done in the past, which only few of them are comparable with humans. The aim of this project, therefore, will be trying to simulate as much as possible humans' capabilities.**

***Index Terms* One-Shot Learning, Genetic Programming, TPOT, Handwritten Characters Recognitions.**

INTRODUCTION

One of the main differences between computers and humans, is that the latter learn from past experiences. Now the question is, can we have computers being able to learn from experience too? The main issue of most of the Machine Learning algorithms is the need of extensive volume of data. On the other hand, the ability to learn object categories from few examples has been demonstrated in humans, [1][2] and it is estimated that a child has learned almost all of the 10 ~ 30 thousand object categories in the world by the age of six [3].

The aim of this paper is to explore a Machine Learning branch, called ***one-shot-learning***, which its goal is trying to classify objects from just one sample, or only few of them. One-Shot Learning was born in computer vision in order to overcome object categorization algorithms that required training on hundreds or thousands of images and very large datasets.

I will present an innovative procedure which, based on limited assumptions of the input model, automatically collects features which enable the classification with a high percentage of success with only few examples.

I will make use of the Omniglot dataset, a collection of handwritten characters from the world's alphabets, and a TPOT classifier, in order to make the machine be capable of learning a large class of visual concepts from just a handful of examples and eventually generalising in ways that are almost indistinguishable from people.

The paper is divided into five sections. The first part concerns a summary of the related works which have been done in the past that have to do with the topic of classification on One-Shot Learning.

The second part introduces the methodology related to what my analysis will achieve; furthermore, it depicts the whole procedure and dataset that will be utilized.

The third section includes a set of experiments I performed by making use of the approach I developed, followed by the results obtained through the method itself.

The fourth part regards a discussion on the results I obtained compared with previous works related to this topic.

The last section infers some conclusion reached by this approach and some comments concerning the whole project.

I. RELATED WORK

Several, but not massive related works have been done in the past regarding this topic which remains, nowadays, "fairly immature" [4].

A few years later the turn of the millennium, Li Fei-Fei et al. in [5], started working on a well-known framework called *Bayesian* which was addressed to unsupervised one-shot learning of object categories. By using this approach, they started with a learned object class model and its corresponding model distribution. Based on a new image, the purpose was to decide whether it contained an instance of their object class.

In 2013, Lake et al. introduced in [6] "*a new computational approach called Hierarchical Bayesian Program Learning (HBPL) that utilizes the principles of compositionality and causality to build a probabilistic generative model of handwritten characters*". Based on the raw pixels of an image, the model was able to search for a "structural description" to label the image itself by a simple combination of its elementary parts and their spatial relations.

Later, Lake et al. went deeply into the subject of One-Shot Learning by dealing with the problem of learning new spoken words which he defined as "*an essential ingredient for language development*" [7]. He made use of a generative Hierarchical Hidden Markov model jointly with a Bayesian inference procedure.

Currently, with the vigorous and rapid grow of the Neural Networks, the most used approach seems to be the Siamese Neural Networks. The latter is a class of neural network architectures that contain two or more identical sub-networks. Identical here means they have the same configuration with the same parameters and weights. Parameter updating is mirrored across both sub-networks [8], [9]. Gregory Koch et al. in 2015, restricted their attention in character recognition; they learned image representations via a supervised metric-based approach with Siamese Neural Networks, then reused that network's features for one-shot learning without any retraining. Others made use of roughly the same approach for face recognition [10] and signature verification [11].

II. Approach

In this project I will train my model by making use of the Omniglot data set: a collection of 1623 hand drawn characters from 50 alphabets. The number of letters in each alphabet varies considerably from about 15 to upwards of 40 characters. For every character there are only 20 examples, each drawn by a different person at resolution 105x105 [12].

For the goal of the final project, I am going to use a Genetic Programming approach. It is an automated method for creating a working computer program from a high-level problem statement of a problem. Genetic programming starts from a high-level statement of “what needs to be done” and automatically creates a computer program to solve the problem. Genetic programming achieves this goal of automatic programming by genetically breeding a population of computer programs using the principles of Darwinian natural selection and biologically inspired operations. For this reason, I will make use of TPOT to learn a classifier based on Random Forest that tries to differentiate between objects being in the same or different class.

TPOT is a Python Automated Machine Learning tool that optimizes machine learning pipelines using genetic programming. It automates the most tedious part of machine learning by intelligently exploring thousands of possible pipelines to find the best one for the data. Once TPOT is finished searching, it provides the Python code for the best pipeline it found [13], [14].

A. Dataset Creating

By having a big number of files, directories and subdirectories as the Omniglot dataset is, I started by converting all the image files contained in the Omniglot folders (32421 images in total) into entries (rows) in CSV files with the following features: *path*, *alphabet*, *character_number*, *image_name* and *class_id*.

The last feature, *class_id*, is the same for images having the same *alphabet* and *character_number*, so that it represents the class which a character belongs to.

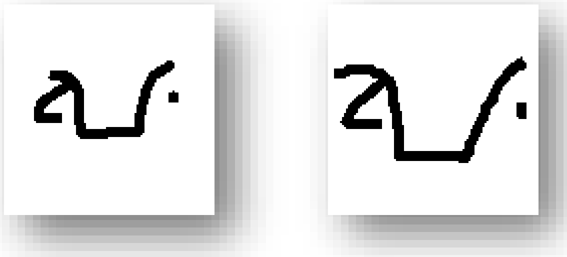
I created, thus, the following datasets as explained in [4] according to Koch:

- **omniglot_dataset.csv**: includes all the image files belonging to Omniglot (32,461 entries);
- **training_set.csv**: includes 30 alphabets out of 50 and 12 drawers for each character out of 20 (11,569 entries);
- **test_set.csv**: includes 10 alphabets out of 20 left and 4 drawers for each character (1,229 entries).

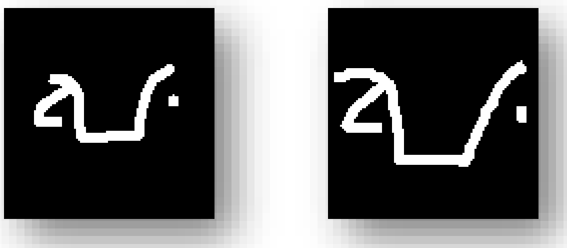
B. Image Processing

My next step was finding a way to compare pairs of images in order to assert whether or not they belong to the same class. To apply this procedure, for every pair of images I used the following approach:

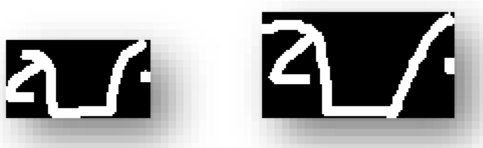
1. Load both images as Image (PIL library) format:



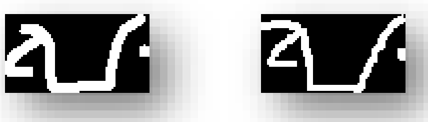
2. Since they are in grey scale, convert every bit of the images (1 minus *pixel_value*):



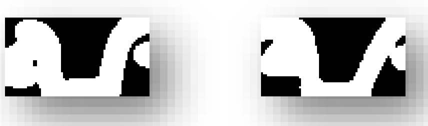
3. Figure out the box containing the object (character) inside the images and crop the images within that box;



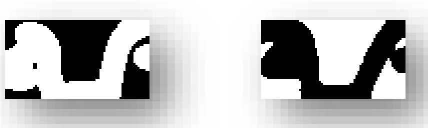
4. Scale the second image with the size of the first one as the objects may have different size:



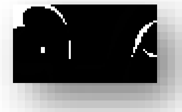
5. Simplify the images by removing (using a filter) useless details that the characters include (contour smoothing, sharpening, etc.):



6. Convert again the grey scale of the second image so that in a pair one image has its object black and the other image has its object white:



7. Overlap the two images to create a new one:



8. Convert the new image into the Omniglot original size (105x105) and then into a Numpy array; finally, take its amount of zeros (number of white pixels remaining) as Machine Learning feature.

By following this approach, images belonging to the same class are likely to have a small amount of zeros

C. TPOT Optimizing and final Classifier

In order to automatically optimise a series of feature pre-processors and models that maximise the cross-validation accuracy on the data set, I made use of the TPOT APIs. Particularly, I generated a classifier based on Random Forest with 5 generations and 50,000 training examples (pairs of images) by sampling random *same* and *different* classes (equally distributed).

At the end of the process, it provided the code for the classifier based on Random Forest with the best Cross Validation score.

I finally used that classifier with again 50,000 training samples to evaluate the performance of the approach for image comparisons.

III. Experiments

To empirically evaluate one shot learning performance, I made use of the same strategy used from Lake et al. in [6]. They evaluated their approach in a 20-Way One Shot Learning tasks for a total of 400 trials. That means, in every trial, the evaluation is made among 20 different classes and only one example: there will be therefore only a pair of characters belonging to the same class and 19 pairs of characters belonging to different classes the one to each other. The goal is to guess as more accurately as possible whether pair of images belong to the same class.

By doing this, my approach reached a value of 92% of accuracy in 400 trials of 20 Way One Shot Learning. In addition, I tested the same method but changing the "*N-Way*"; I repeated, therefore, the test for 10 Way and 5 Way One shot Learning obtaining roughly the same accuracy achieved for 20 Way.

IV. Discussion

Table 1 [4] depicts the best 20 Way One-shot accuracy from each type of approach. However, the same table clearly illustrates how humans have not been surpassed yet. By having a high percentage of 95.5%, in fact, the only two methods out of 9 that seem to be properly compared to humans capabilities are Convolution Siamese Networks with 92%, and even better Hierarchical Bayesian Program Learning with 95.2%. My approach, in this ranking is located at the third place jointly with Convolution Siamese Networks.

METHOD	TEST
1-Nearest Neighbor	21,7
Simple Stroke	35,2
Siamese Neural Net	58,3
Deep Boltzmann Machine	62
Hierarchical Deep	65,2
Affine mode	81,8
Convolutional Siamese Net	92
Paris' Approach	92
Hierarchical Bayesian Program Learning	95,2
Humans	95,5

Table 1 – 20 Way *One-Shot Accuracy*

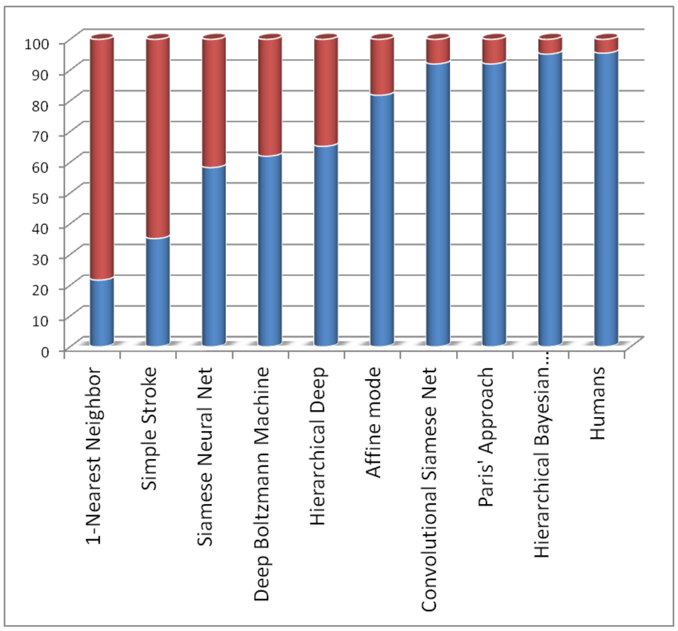


Diagram 1 – 20 Way *One-Shot Accuracy*

V. Conclusion

I have presented in this paper a Machine Learning method based on Genetic Programming which makes use of TPOT, a Python Automated Machine Learning tool. The approach I developed is able to perform classifications from only a small set of samples. Particularly, the procedure has been tested in the so called 20 Way One Shot Learning where there is only one example out of 20.

The method has been performed for handwritten characters recognition by using the Omniglot Dataset. By having, in fact, a small set of character samples, it will be able to recognize whether pairs of random characters written by different people are actually the same character.

This method appears to be in the first three positions as percentage of accuracy, as illustrated in Table 1 and Diagram 1.

By developing this project, I gradually increased my experience and knowledge on Machine Learning techniques, especially by classifying samples with different classification methods and/or different parameters. In addition, I discovered the powerful and the beauty of Automated Machine Learning tools, as TPOT is.

In conclusion, as shown in Table 1, it has not been developed yet any approach that is as smart as humans. However, by studying further some details of the dataset, the comparison may become thinner.

References

[1] F.F. Li, R. VanRullen, C.Coch, and P. Perona, "*Rapid natural scene categorization in the near absence of attention*". PNAS, 99(14):9596-9601, 2002.

[2] S. Thorpe, D. Fize, and C. Marlot, "*Speed of processing in the human visual system*". Nature, 381:520-522, 1996.

[3] I. Biederman. "*Recognition-by-Components: a theory of human understanding*". Psychological Review, 94:115-147, 1987.

[4] Koch, G. R.. “*Siamese Neural Networks for One-shot Image Recognition*”, 2015.

[5] Fe-Fei, Li, Fergus, Robert, and Perona, Pietro. “*A Bayesian approach to unsupervised one-shot learning of object categories*”. In Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, pp. 1134–1141. IEEE, 2003.

[6] Lake, Brenden M, Salakhutdinov, Ruslan R, and Tenenbaum, Josh. “*One-shot learning by inverting a compositional causal process*”. In Advances in neural information processing systems, pp. 2526–2534, 2013.

[7] Lake, Brenden M, Lee, Chia-ying, Glass, James R, and Tenenbaum, Joshua B. “*One-shot learning of generative speech concepts*”. Cognitive Science Society, 2014.

[8] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Sickinger and Roopak Shah. “*Signature Verification using a Siamese Time Delay Neural Network*”. AT&T Bell Laboratories, 1994.

[9] YIN, Wenpeng et al. “*ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs*”. Transactions of the Association for Computational Linguistics, [S.l.], v. 4, p. 259-272, jun. 2016. ISSN 2307-387X.

[10] Firdaouss Doukkali, “*One-Shot Learning: Face Recognition using Siamese Neural Network*”. Toward Data Science. Internet: towardsdatascience.com/one-shot-learning-face-recognition-using-siamese-neural-network-a13dcf739e.

[11] Sounak Dey, Anjan Dutta, J. Ignacio Toledo et al. “*SigNet: Convolutional Siamese Network for Writer Independent Offline Signature Verification*”. 2017

[12] Omniglot, The online encyclopedia of writing systems and languages. Internet: http://www.omniglot.com/.

[13] Randal S. Olson, Ryan J. Urbanowicz, et al. "*Automating biomedical data science through tree-based pipeline optimization*". Applications of Evolutionary Computation, pages 123-137. 2016.

[14] Randal S. Olson, Nathan Bartley, Ryan J. Urbanowicz, and Jason H. Moore. "*Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science*". Proceedings of GECCO 2016, pages 485-492, 201.