Face/Comic recognizer

Matteo Turla *

June 2022

Abstract

The goal of the project is to implement a deep-learning based system discriminating between real faces and comics.

1 Introduction

The goal of the project is to implement a deep-learning based system discriminating between real faces and comics. To do so it has been tuned and trained a small fully-connected neural network achieving 99.7% accuracy on the test set. Section 2 gives an introduction of the chosen dataset and how it has been organized, Section 3 explains the applied pre-processing techniques and the structure of the model, in Section 4 it is explained the used algorithm to optimize the model and how the algorithm scale up with the size of the data and finally in Section 5 are described the experiments and the results.

2 Dataset

The *Comic faces* by *Alex Spirin* [1], published on Kaggle, is composed of 20k labeled *real vs comic* images and it can be used to perform binary classification of images into two labels: **real face** and **comic face**. The dataset is organized into two folder, one for each label, named **faces** and **comics**. Each folder contains 10k images. To train, validate and test the neural network it has been used the train-validation-holdout strategy: for each label it has been considered the first 5k images as training set, the next 2k images as validation set and the final 3k images for the holdout set. A more in depth explanation on how this

*I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

[1] https://www.kaggle.com/datasets/defileroff/comic-faces-paired-synthetic-v2

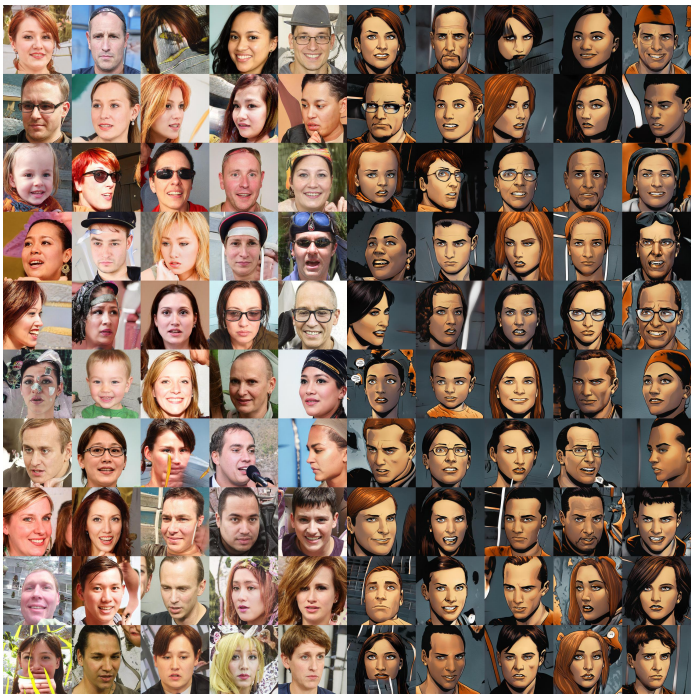

Figure 1: Sample of the dataset

splits are used is given in Section 5, note that the holdout set is only used to estimate the accuracy of the final model and it is not used to take decisions about the hyperparamters of the learning algorithm and decision about the architecture of the neural network.

Figure 1 shows a subset of images present in the dataset.

3 Preprocessing and Neural Network architecture

The images are first scaled to 24x24 pixel, then each RGB channel is normalize with the following parameter $mean = (0.485, 0.456, 0.406)$ and $sdv = (0.229, 0.224, 0.225)$ and finally they are flattened to a $24 \times 24 \times 3$ vector. The reason of the scaling to a such small resolution is due to the fact that the main characteristics used to discriminate between real and comic faces is the color and not the level of details. Reducing the images to a small scale allow the network to be smaller and so it is faster to train and to evaluate.

The chosen model is a fully connected neural network composed of 3 layers, it is used the ReLU activation function and no regularization layer are present in the architecture. The model can be described with the following function

$$y = softmax(C \cdot ReLU(B \cdot ReLU(Ax)))$$

where A, B, C are learnable matrix of size $2 \times 128, 128 \times 128, 128 \times 1728$ respectively, x is the input vector of size 1728 and ReLU is the activation function $ReLU(z) = max(0, z)$.

4 Algorithm

The used algorithm to optimize the learnable parameters of the neural network is called stochastic gradient descent. SGD use the following rule in order to update the learnable parameters:

$$w_{t+1} = w_t - \eta \times \frac{\mathrm{d}L}{\mathrm{d}w}(w_t)$$

Where η is the learning rate and L is the chosen loss function to minimize.

To compute the derivatives of the loss function with respect to the learnable parameters and update them it is required to have in main memory all the parameters of the neural network and the training examples. SGD only use a subset of training examples called batch. Thus it is needed to load in main memory only the examples belonging to the batch while all the other can be stored on the secondary memory. The number of parameters of a state of the art neural network are of the order of 100×10^6, this only requires $100 \times 8 \times MB$, while the size of the examples to load in main memory depends on the batch size B: $B * 1728 * 8Bytes$. Thus SGD by using only a subset of the data at each iteration, it is able to scale the learning procedure to a huge number of training examples without increasing the memory complexity.

5 Experiment

Some of the most important hyperparameters in training a neural network are the *learning rate* and the number of *epochs*. By tuning these parameters is possible to find the right balance between overfitting and underfitting. In order to tune them is has been used a grid search strategy: the neural network is trained on the training set and tested on the validation set for each combination of the hyperparameters. Then the best combination of hyperameters is used to train the neural network over the union of the training and validation set. To estimate the performance of the model on unseen data it is used the holdout set, which has never been used before.

The used loss function is called *Cross-entropy loss function* and the used batch size is 64. the tuned parameters are:

1. learning rate: 0.01, 0.001
2. number of epochs: 1, 3, 5

6 Results

The best combination of hyperparameters, shown in the table below, is given by a learning rate of 0.001 and 3 number of epochs. The estimate of the accuracy computed over the holdout set is 0.997.

Validation accuracy | loss			
Grid	n epochs: 1	n epochs: 3	n epochs: 5
l.r. 0.01	0.987 | 0.049	0.996 | 0.01	0.996 | 0.012
l.r. 0.001	0.996 | 0.014	0.997 | 0.005	0.995 | 0.015

7 Conclusion

In this project it has been studied and implemented a fully connected neural network to perform binary classification. Even if a fully connected neural network is not the best choice when working with images, it correctly classified 99.7% of the images in the holdout set.