IDs:

ID #1: 300822954

ID #2: 307963538

Deep Learning

Assignment 2 – Siaemese Neural Networks and one shot learning.

As instructed, our

**We use the following base configuration:**

**Experiments:**

The models were obtained by a training procedure which was bounded by

Also, an exponential learning rate decay mechanism has been applied, i.e., .

We stored the results including the training log, which includes the number of iterations and epoch, batch size, regularization method, training accuracy score per epoch, validation accuracy per epoch, and test set accuracy score.

We trained various models based on the following grid-search, with hyper-parameters as follows:

**Section 2C – EDA:**

We started our EDA by verifying that no subject )which we didn’t know whether it refers to pairs or individuals) is shared between the training and testing sets, as stated in the instructions. This was verified to be true for individuals and, as a an outcome, for pairs as well. From visually inspecting the image files, we also noticed that aside from centering and cropping the images to contain the individuals’ faces, some images were also cropped and rescaled, adding a completely black background in the edges of the frame.

To asses the scale of the provided image set, we checked how many image files each individual has in the image (‘Ifw2Data’) directory supplied and plotted the histogram of the individual counts of image files (Fig. 1). We discovered that the majority of individuals have only a single image file, while some individuals have hundreds of image files, with the maximum amount saved under *‘George\_W\_Bush’* name, with 530 images.

Chart, histogram

Description automatically generated

Figure 1:Distribution of Image files count - limit of y-axis is set to 10. Maximum count of image files detected is 530.

We then focused on the training and testing set only, disregarding any finding related to the image files themselves, as we are instructed to train and test our data using only the images addresed by the ‘pairDevTrain.txt’& ‘pairDevTest.txt’; We found that the individual appearing in the training set and in the test set are ‘Alec\_Baldwin’ ‘Tang\_Jiaxuan’ , accordingly, both with 6 appearances only (both under matching pairs and non matching pairs). We also plotted a histogram of all individuals’ amount of appearances in the training and testing sets (Fig. 2), and although most individuals appeared once, it seems there is some sort of a power law restricting the amount of duplicated (or more) inviduals’ appearances in both training and testing sets. In the training set, only 351 individuals appeared in both the matched pairs and non matched pairs, whilst 138 appeared in both matching and non matching pairs in the testing set.

A picture containing icon

Description automatically generated

Figure 2 - Distributions of appearances in training and testing sets. The total number of unique individuals in the training set is 2132, the number of unique individuals in the matching pairs of the training set is 788, and in the non matching pairs 2132. The total number of unique individuals in the testing set is 963, the number of unique individuals in the matching pairs of the testing set is 353, and in the non matching pairs 748.

Checking for shared individuals across matching and non matching pairs is not sufficient, as an individual might be represented by a different image (notated with the image index) in the matching samples then in the non matching samples. Hence, we checked whether the set of images used for the matching samples is fully/disjoint/contiguous with the set of images used for the non-matching sample with all shared individuals. We found that all the images of individuals appearing in both matching and non matching pairs are the same in the matching and non matching samples.

EDA conclusions:

As most individuals appear in either the matching pairs **or** the non-matching pairs, the task in hand (one-shot learning to classify if two images are of the same individual) is more difficult than we thought. During most of the training, our model will learn from each individual only once, either from matching samples or non-matching samples, and will have very few opportunities to learn matching and non matching samples of a single person. This makes the task of training closer to zero-shot learning. This is emphasized by the fully disjoint train and test sets, were no individuals are contiguous about the two sets.

**Pre-processing steps:**

**Models’ Network Architecture:**

As specified in the assignment, each model

**Siaemese NN implementation details:**

We followed the

Stopping criterion:

**Empirical results:**

Accuracy plots:

batch\_size=16 batch\_size=32 batch\_size=64

**The effect of the different batch sizes on training performance/convergence**

In this experiment, we look at the effect of using different batch size values on model performance/convergence.

**Results:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Batch\_size | Epoch | Iterations | Train accuracy | Test accuracy |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

As can be seen in the table above,

Note:

**The effect of the different learning rate on training performance/convergence**

**Remarks, Observation, and Comparison:**

1. From our experiments when

**Conclusion:**

The

**How To Run:**

1. open a terminal and cd to ‘HW1\_300822954\_307963538’ directory

2. conda create --name HW1\_300822954\_307963538 python=3.8

3.WINDOWS: activate HW1\_300822954\_307963538

LINUX, macOS: source activate HW1\_300822954\_307963538

4. pip install -r requirements.txt

For running our code, run

Default configuration: