Deep Learning assignment 2:

Siamese Networks for One-Shot Learning, evaluated with Labeled Faces in the Wild data set:

Analysis of the dataset:

Size: The dataset includes 13,233 images of 5749 people.

Number of examples:

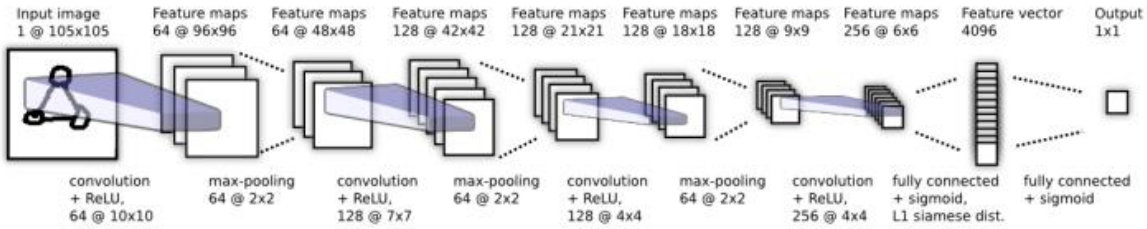
- The training set include 2200 Couples of images – the first 1100 are for the same person (**class 0**), the rest 1100 for different persons (**class 1**).
- The test set include 1000 Couples of images – the first 500 are for the same person (**class 0**), the rest 50 for different persons (**class 1**).
- We shuffled the train set randomly and split it such as 15% of it will move to a validation set.

Experimental setup:

Batch size – 32

Parameters of architecture:

Our network's architecture is almost as in the referenced paper:



---

4 convolution layers, after each of them was relu activation, 2 fully connected with sigmoid activation:

Layers architecture (by their position):

1. Convolutional layer with 64 filters of 10*10 followd by relu activation, and 2*2 max pooling
2. Convolutional layer with 128 filters of 7*7 followd by relu activation, and 2*2 max pooling
3. Convolutional layer with 128 filters of 4*4 followd by relu activation, and 2*2 max pooling
4. Convolutional layer with 256 filters of 4*4 followd by relu activation, and 2*2 max pooling
5. Fully connected layer with 4096 neurons followd by sigmoid activation
6. Fully connected layer with 1 neuron followd by sigmoid activation to predict similarity.

Forward propagation flow:

For input of 2 images, for each image separately go in the convolutional layers and the first fully connected.

At this stage we should have two tensors of size 4096,1.

Now calculate l1 distance between the two tensors, and send the result to the last fully connected layer which supposed to predict their similarity.

Learning rates – 0.01 with changed decay (will explained later, in the evaluations)

optimization – Adam optimizer with weight decay (L2 penalty) of 1.

Batchnorm – for each convolutional layer

Stopping criteria – stop after the validation error didn't decrease for 3 epochs (in the paper it was 20 but our model converges much faster) and take the best parameters (the epoch with the lowest validation error)

Reason for those parameters – referenced from the paper, we will try changed combination or the scheduler parameters and present the difference.

Resize images to 105*105 because without it the session crashed again and again due to ram memory explosion.

Analysis of architecture's performance for different scheduler parameters:

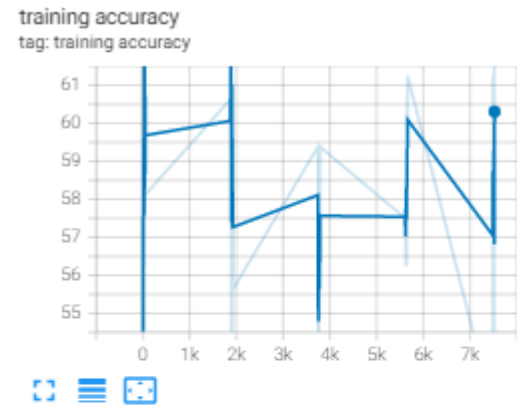
First experiment: 'StepLR' scheduler for learning rate (**step_size** = 1 which - Multiply the learning rate of each parameter group by the factor given in the specified function)

a. Convergence times – 2.5 min
b. final loss and accuracy on the holdout set– (0.767, 58.1%)) respectively
c. final loss and accuracy on the test set – (0.75, 58.1%) respectively
d. Graphs describing the loss on the training set throughout the training process

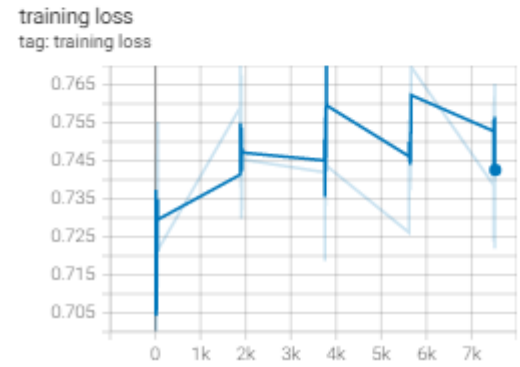
Second experiment: 'MultiStepLRscheduler' scheduler for learning rate (**milestones** = [1,3,5] which - Decays the learning rate of each parameter group by gamma once the number of epoch reaches one of the milestones. Notice that such decay can happen simultaneously with other changes to the learning rate from outside this scheduler.)

a. Convergence times – 2.5 min
b. final loss and accuracy on the holdout set– (0.742, 61.1%)) respectively
c. final loss and accuracy on the– test set (0.75, 57.7%) respectively
d. Graphs describing the loss on the training set throughout the training process

training accuracy



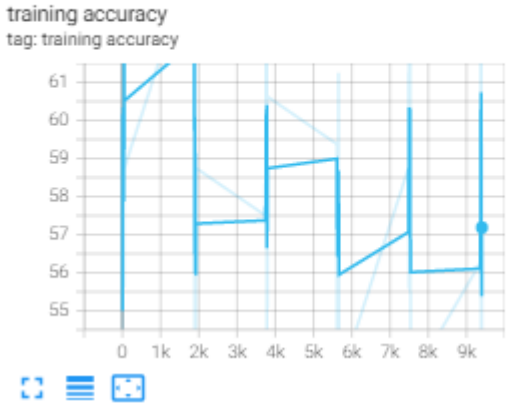
training loss



Third experiment: 'ExponentialLR' scheduler for learning rate (**gamma**= 0.1 which - Decays the learning rate of each parameter group by gamma every epoch.)

a. Convergence times – 3 min
b. final loss and accuracy on the holdout set (0.781, 41.2%)– respectively
c. final loss and accuracy on the test set– (0.747, 61.5%) respectively
d. Graphs describing the loss on the training set throughout the training process

training accuracy

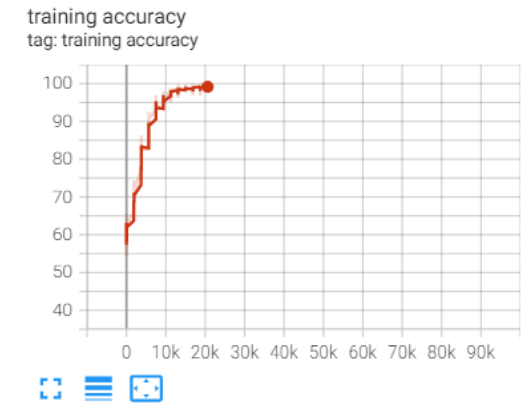


training loss

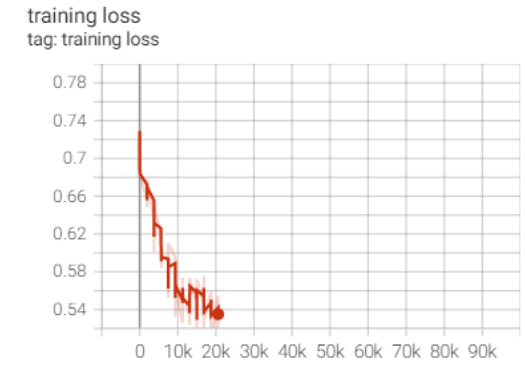
Fourth experiment: eliminate scheduler and decrease learning rate to be 10^-4.

That brings the best performance

a. Convergence times – 1.5 min
b. final loss and accuracy on the holdout set (0.538, 99.2%)– respectively
c. final loss and accuracy on the test set– (0667, 78.5%) respectively
d. Graphs describing the loss on the training set throughout the training process

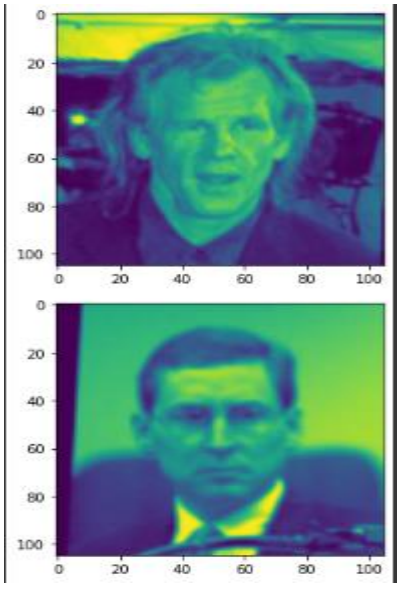


training loss

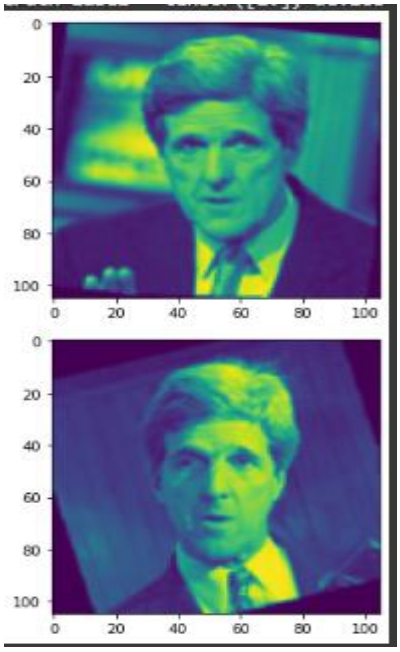


Examples of accurate and misclassifications and trying to determine why the model was not successful:

Success (real label = not the same person) –



Misclassification (real label = the same person) –



It can be seen that the success was for different person with Different facial features (one has a round face and the other has thin faces), different hair, different forehead height and more...

When the misclassification was for the same person – the image was augmented,maybe adding an augmentation could improve the model.