ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING - PROJECT REPORT

Project Report on Siamese Networks for glyphs

By: Drashti Kanubhai Nayak (n10599568)

This project report is submitted to the Science and Engineering faculty in partial fulfilment for the Master of Information Technology degree.

The report describes the working of a deep neural network classifier to predict whether two images fall in the same class and states the experimental results of the classifier.

Table of Contents:

Introduction:
Siamese networks are unlike normal classification which requires a huge amount of data images. Object detection based on image recognition is not possible with the normal classification method because for real-time detection, and one cannot get thousands of images to learn from in real-time. Siamese network learns through single-shot image learning. Even if a new image arrives it will compare and learn. The drawback of the siamese network is its learning time which is quite high.
We have to use contrastive and triplet loss in creating a deep neural network siamese network to identify two images and their classes. The process for creating a siamese network has been the same in both files except measured based on different loss functions i.e. Contrastive loss function and Triplet Loss Function. The contrastive loss function will take the output and measure distance with the same class based on positive and negative examples. Triplet loss function also embeds based on similarity and dissimilarity compared with a positive and negative example; the only difference is triplet does not deal with anchor-neighbour pairs of samples whereas contrastive does. If positive examples are compared with a more similar class; the loss will be low and negative will showcase distant sample examples. The loss explains the prediction capability between both functions.

Architecture:
The architecture of siamese networks is similar for both loss functions. In the training phase, the input of the model will consist of two images for learning. In both images, the model will find embedding vectors viz. Feature extraction of two images separately and will calculate the difference in the last layer and the value of difference will be stored. Hence, when next time an image detection arrives it can compare it with a new image, detect it and further classify. The concept of face reidentification using siamese networks can further enhance the concept.

Dataset: Dataset consists of sets of classes consisting of different language handwritten letters. The dataset is useful for human-like learning. It consists of 1623 characters and 20 examples.
In Input, the data will go in pairs and it can be similar or different i.e. handwritten letters can be the same letter or different based on different language classes. This input also has the label of similarity or similarity i.e. its difference.

Code:
Contrastive Loss: In the deep neural network, we have used 3 Max Pooling layer with activation function = Relu and the Last flatten layer is used with a sigmoid activation function.

Triplet Loss same as Contarastive loss. Then, added a customized layer to compute the absolute difference between the encodings. Later, added a dense layer with a sigmoid unit to generate the similarity score. In the end, we connect input with output values and return the values from the model.

The siamese model summary for Contrastive loss:

```
model = get_siamese_model((105, 105, 1))
model.summary()
```

```
Model: "model"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, 105, 105, 1) 0
__________________________________________________________________________________________________
input_2 (InputLayer)            [(None, 105, 105, 1) 0
__________________________________________________________________________________________________
sequential (Sequential)         (None, 4096)         38947648    input_1[0][0]
                                                                 input_2[0][0]
__________________________________________________________________________________________________
lambda (Lambda)                 (None, 4096)         0           sequential[0][0]
                                                                 sequential[1][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 1)            4097        lambda[0][0]
==================================================================================================
Total params: 38,951,745
Trainable params: 38,951,745
Non-trainable params: 0
__________________________________________________________________________________________________
```

The siamese model summary for Triplet loss:

```
[21] model = get_siamese_model((105, 105, 1))
     model.summary()
```

```
Model: "model"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, 105, 105, 1) 0
__________________________________________________________________________________________________
input_2 (InputLayer)            [(None, 105, 105, 1) 0
__________________________________________________________________________________________________
sequential (Sequential)         (None, 4096)         38947648    input_1[0][0]
                                                                 input_2[0][0]
__________________________________________________________________________________________________
lambda (Lambda)                 (None, 4096)         0           sequential[0][0]
                                                                 sequential[1][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 1)            4097        lambda[0][0]
==================================================================================================
Total params: 38,951,745
Trainable params: 38,951,745
Non-trainable params: 0
__________________________________________________________________________________________________
```

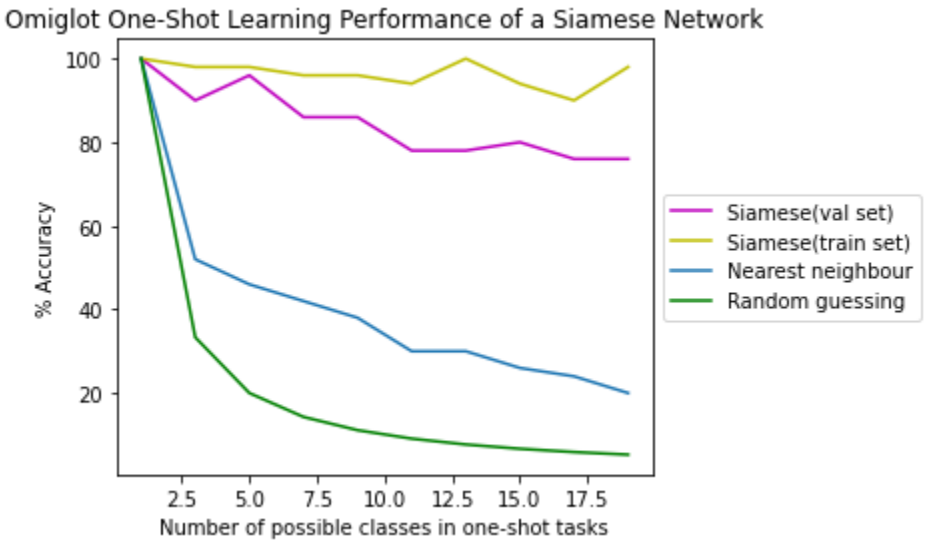
✓ 0s completed at 10:43 PM

So, we prepared the data as per the required inputs. Hence we gave input to the siamese network, created the model, gave optimiser, gave loss and after training we tested them.

3. Results:

Compared to contrastive loss, the performance measure of triplet loss is quite low. Hence, based on this assignment requirement, when hyperparameters have been kept the same the contrastive loss function works better than the triplet loss function. If we tune it different than contrastive then triplet might produce a better results.

Contrastive loss:



Triplet Loss: