Graphical user interface, text

Description automatically generated

IFN680 ASSIGNMENT TWO REPORT

Siamese Network

**Student Name**: Ajay Moktan

**Student ID**: n10349561

Lecturer: Frederic Maire

Tutor: Will Browne

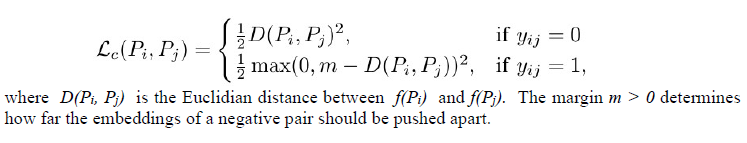
**Introduction**

The neural network can be trained to learn similarity relations between objects. The main idea is to learn an embedding function such that similar objects are mapped to close points and non-similar objects are mapped to points that are apart at a distance. This Convolutional Neural Network is called Siamese neural network because they are used in two copies of the same network on two different input vectors. The Siamese networks are used for recognizing handwritten checks, automatic detection of faces in camera images, animal in the wild re-identification, and matching queries with indexed documents (Frederic, 2021). Thus, this report will use an Omniglot dataset for the experiment which is a collection of hand-drawn characters from different alphabets. Moreover, it will use the two embedding functions which are contrastive loss function and triplet loss function. Lastly, the two embedding functions will be trained to the built Siamese network and evaluate the performances on the training set, testing set, and both sets.

**Siamese Network**

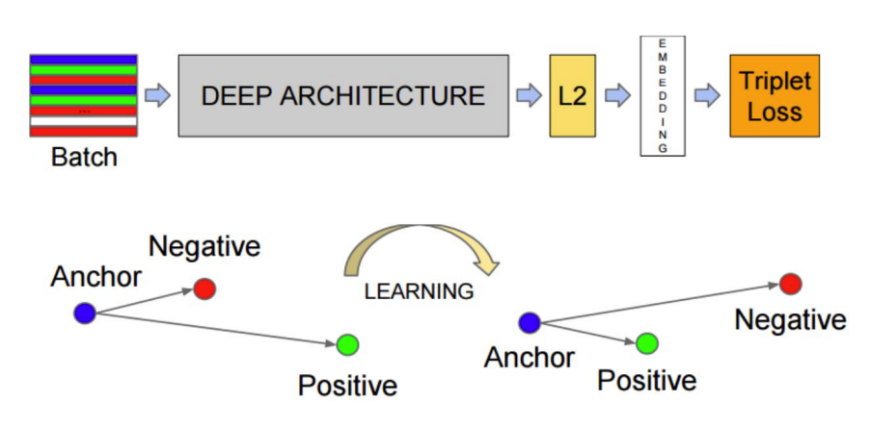
Siamese Network which can be called a twin neural network is a convolutional neural network that has **two identical networks.** This means the identical networks share the same weights and parameters. It works by training one of the subnetworks and using the same configuration for other subnetworks as well. These are used to find the similarities of the inputs with their comparison of feature vectors. In specifically, the networks are trained to maximize the contrast distance between the embedded inputs of different classes. Oppositely, it minimizes the distance between embeddings of similar classes. Thus, the distance becomes shorter as it is similar which is positive and the distance becomes larger if they are different which is negative. **This is trained by the loss functions called contrastive loss function and Tiplet loss function.**

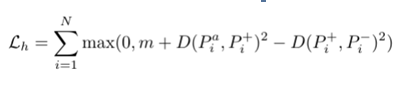
**Contrastive Loss Function**

Contrastive loss is margin-based (**trying to increase the margin between dissimilar pairs while decreasing between similar pari)** for a two steam network. Thus, the training of a Siamese network is done with a pair of positive and negative objects with its label. This is performed by optimizing a contrastive loss function. The equation of contrastive loss functions is the following.

**Triplet Loss Function**

Unlike contrastive loss, triplet loss function takes three inputs; anchor, positive and negative. Positive is the one similar to the anchor while negative is dissimilar to an anchor. It attempts to learn a ranking of similarities and enables positive example should be closer to the 'anchor' while negative is pushed farther away. In other word, negative examples will have a long-distance from anchor example than positive example. The following figures will show how the triplet loss function works with an equation



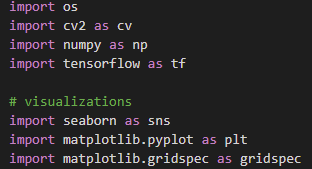


**Omniglot dataset**

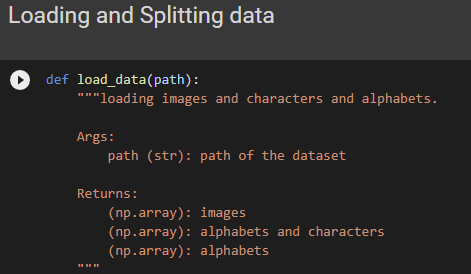
Omniglot dataset is a collection of 1623 hand-drawn characters from 50 different alphabets. For every character, there are just 20 examples, each drawn by a different person. Each image is a greyscale image of resolution 105x105. For the experiment, the datasets are formatted into two folders named images\_background for training purposes and images\_evaluation folder for testing purposes.

**Experiment**

* Importing the required modules for the experiment:

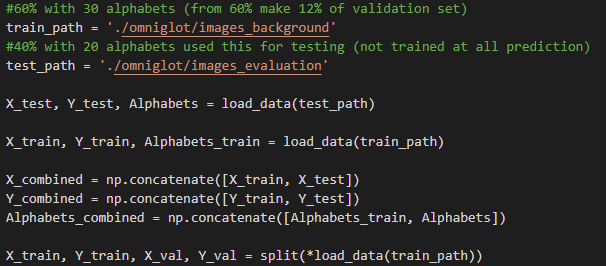


* Loading the **Omniglot** dataset from Tensor Flow

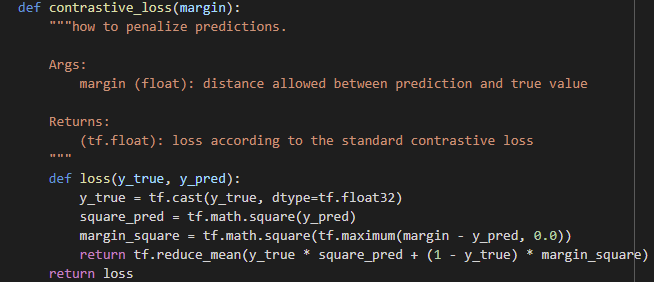


The Omniglot dataset couldn’t download directly in the environment. Therefore, the dataset was downloaded manually from the official online source.

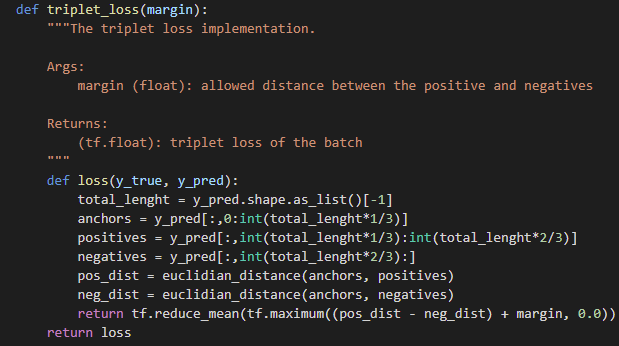
* Splitting the dataset into **train** and **test** set and **both** set



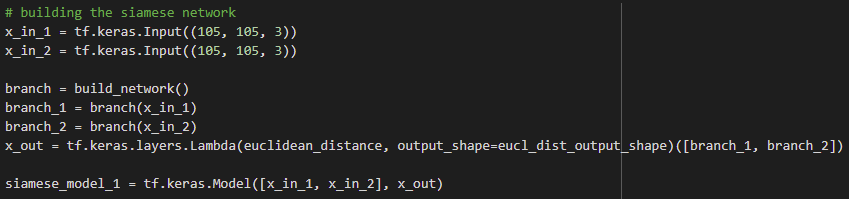
* Implementing **Contrastive loss function**



* Implementing **Triplet loss** function



* Build a Siamese network

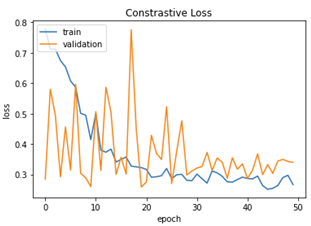


Before building the Siamese network the CNN was built for the base. There are two models of the Siamese network based on one CNN which is for contrastive loss with two branches and triplet loss with three branches.

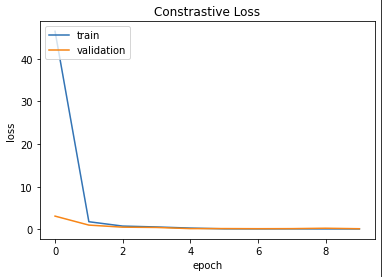
* Train the built Siamese network on the **training** set. Plot the training and **validation error vs time**

Evaluate the performance of the network trained with two loss functions:

**Contrastive loss function:**



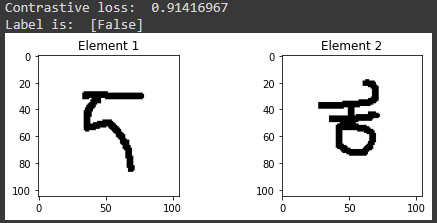
Referring to the above plot graph, using the implemented contrastive loss function on the built Siamese network shows that model is learning but there are overfitting identified as well. Thus, the regularizations L2 were added and edited the learning rate of Adams in optimizer into the model to avoid overfitting.



By decaying the weights with L2 regularization, the plot clearly shows that overfitting has decreased and the two lines are converging effectively with good learning in the model.

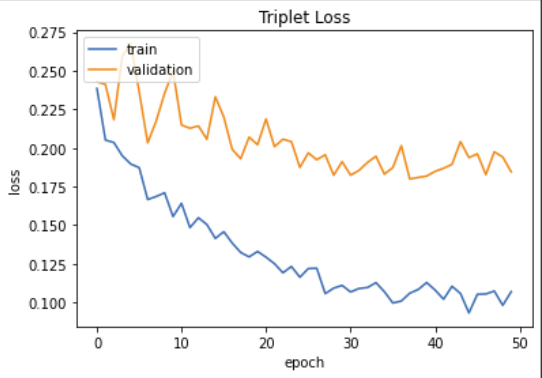
|  |  |  |
| --- | --- | --- |
| **Dataset** | **Epochs** | **Loss/error** |
| Train set | 10 | 0.1200 |
| Both set | 10 | 0.1201 |
| Test set | 10 | 0.1211 |

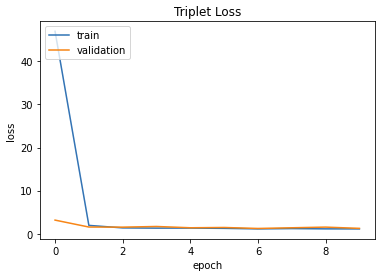
The above table is the results for 3 splits using contrastive loss functions which are train, combined, and test sets. Comparing the 3 sets the test set has the highest errors which make sense as the test sets have never been seen and trained in the model. However, the 3 sets have similar results between each other which are a good signal that the Siamese network is training and learning sufficiently which means it is generalized capable.



The following screenshot is a printing of one of the false predictions from the Siamese network using contrastive loss in test sets.  The results of 0.914 can see as a distance. As the distance is higher there is a higher chance of being a false alphabet to element 1. Thus, in this case, as the loss is lower or very close to 0 there is a high chance of those two elements being the same alphabets.

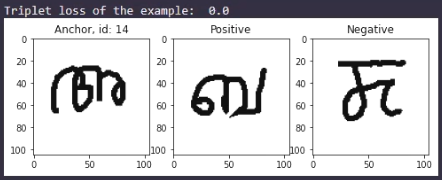
**Triplet loss function:**



Same with the triplet loss function before adding the L2 regularization there is overfitting. However, adding the L2 regularization and editing the learning rate of Adams to the model has solved the overfitting dramatically as below in the plot chart.

|  |  |  |
| --- | --- | --- |
| **Dataset** | **Epochs** | **Loss/error** |
| Train set | 10 | 0.9016 |
| Both set | 10 | 1.8225 |
| Test set | 10 | 2.7285 |

The above table is the results for 3 splits using the triplet loss function which are train, combined, and test sets. Comparing the 3 sets the test set has the highest errors which makes sense as the test sets have never been seen and trained in the Siamese network. As the purpose of the Siamese network is generalizing and also predicting unknown classes the results shown from the table are reasonable. Therefore, it means the model was capable.



The following screenshot is for qualitative analysis, it’s one of the false predictions from the Siamese network using triplet loss functions in the test set which has an anchor, positive and negative. In this case, there are two terms in triplet loss that the results are displaying. It’s making a mistake in positive or negative. For example, referring to a false prediction tells that the anchor is very different from positive or the anchor is very similar to negative. Thus, the number is a sum of the differences between negative and positive. However, the above results have triplet loss of the example as **0** which means the positive is the same alphabet to anchor and the negative is dissimilar to the anchor correctly.

**Conclusion**

In conclusion, this experiment has successfully built Siamese networks for the Omniglot dataset predicting the alphabets using two loss functions which are contrastive and triplet. The ability of the Siamese network to generalize on the dataset they were trained on, is tested against the alphabets it is not necessarily been exposed to. Moreover, the generalisation capability of Siamese network enables the network to recognise the similar objects they were once trained and hence, does not require re-training for every time differently written character are fed into the network. However, there are some limitations between the results on the training set and the validation set which indicates overfitting. To improve the generalization, the regularization has been added to the model such as L2 to decay the weights and edited the learning rate of Adams. Nevertheless, many hyperparameters can be tuned in order to avoid overfitting which needs further works in the future.

**Reference list**

G. (2020, October 19). One Shot Learning and Triplet Loss. Kaggle. <https://www.kaggle.com/gawarek/one-shot-learning-and-triplet-loss>

G. (2020, October 19). One Shot Learning and Triplet Loss. Kaggle. <https://www.kaggle.com/gawarek/one-shot-learning-and-triplet-loss>

Team, K. (2021). Keras documentation: Image similarity estimation using a Siamese Network with a contrastive loss. Keras. <https://keras.io/examples/vision/siamese_contrastive/>