Siamese Neural Network for One-shot Image Recognition

Sankhla Surabhi
Yadav Saumya

Introduction:

While deep neural networks have proved to be successful at learning on really complex data sets, but they almost always need large training datasets of the same object to be effective. For example, a large dataset of many labelled cat and non-cat images is required for the neural network to learn how to identify a cat in an image. A human on the other hand is very likely to identify that something is a fruit even if seen just once because they have grown up seeing multiple fruits. This is called *one shot learning.*

Why is one shot learning important?

In many cases, the data available from the given class is limited and that makes it difficult to do recognition or verification tasks. In this case, we can train a one shot learning model on other similar classes where we have a lot of data. For example, if we want to build a face verification system of a company, we might have just one image per employee which makes a neural network quite redundant. However, we can train a one shot learning model on a huge dataset of faces and then use this employee image for verification.

Related Work

In our project we implement the siamese network described in work by *Koch et. al. of Department of Computer Science, University of Toronto Learning* on AT&T dataset of faces.

The paper uses a Siamese Neural network for one shot learning. A siamese neural network consists of twin convolutional networks which accept distinct inputs but are joined by an energy function at the top.

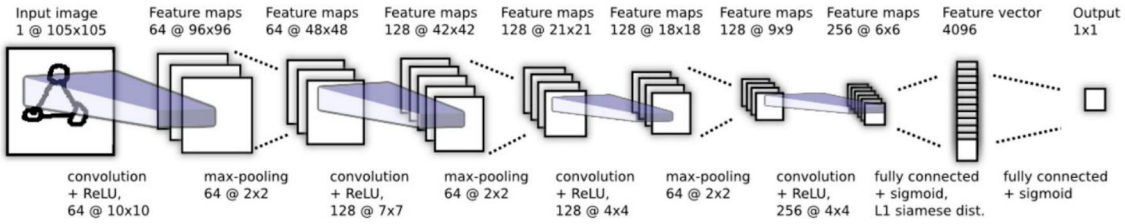
Architecture and loss function

The implementation in this research paper uses the weighted L1 distance between the twin feature vectors h1 and h2 combined with a sigmoid activation, which maps onto the interval [0, 1]. Thus a cross-entropy objective is a natural choice for training the network. They use multiple convolutional layers before the fully-connected layers and top-level energy function.

The loss function is given as:

$$\mathcal{L}(x_1^{(i)}, x_2^{(i)}) = \mathbf{y}(x_1^{(i)}, x_2^{(i)}) \log \mathbf{p}(x_1^{(i)}, x_2^{(i)}) + \\ (1 - \mathbf{y}(x_1^{(i)}, x_2^{(i)})) \log (1 - \mathbf{p}(x_1^{(i)}, x_2^{(i)})) + \boldsymbol{\lambda}^T |\mathbf{w}|^2$$

The image below shows the architecture of one of the twins. 3 Blocks of Cov-RELU-Max Pooling are used followed by a Conv-RELU connected to a fully-connected layer with a sigmoid function. This layer produces the feature vectors that will be fused by the L1 weighed distance layer. The output is fed to a final layer that outputs a value between 1 and 0 (same class or different class).



We used one of the open source implementation of this neural network (enclosed here) as a starting point.

Dataset

We started our project with the same dataset as the research paper, i.e., the omniglot dataset. This dataset contains 1623 different handwritten characters from 50 different alphabets. Each of the 1623 characters was drawn online via Amazon's Mechanical Turk by 20 different people. However, as we trained the github neural network on this dataset, we found it too computationally intensive for the purpose of the project. It took 105 mins for 16000 iterations and reached an accuracy of 55.4% on validation set.

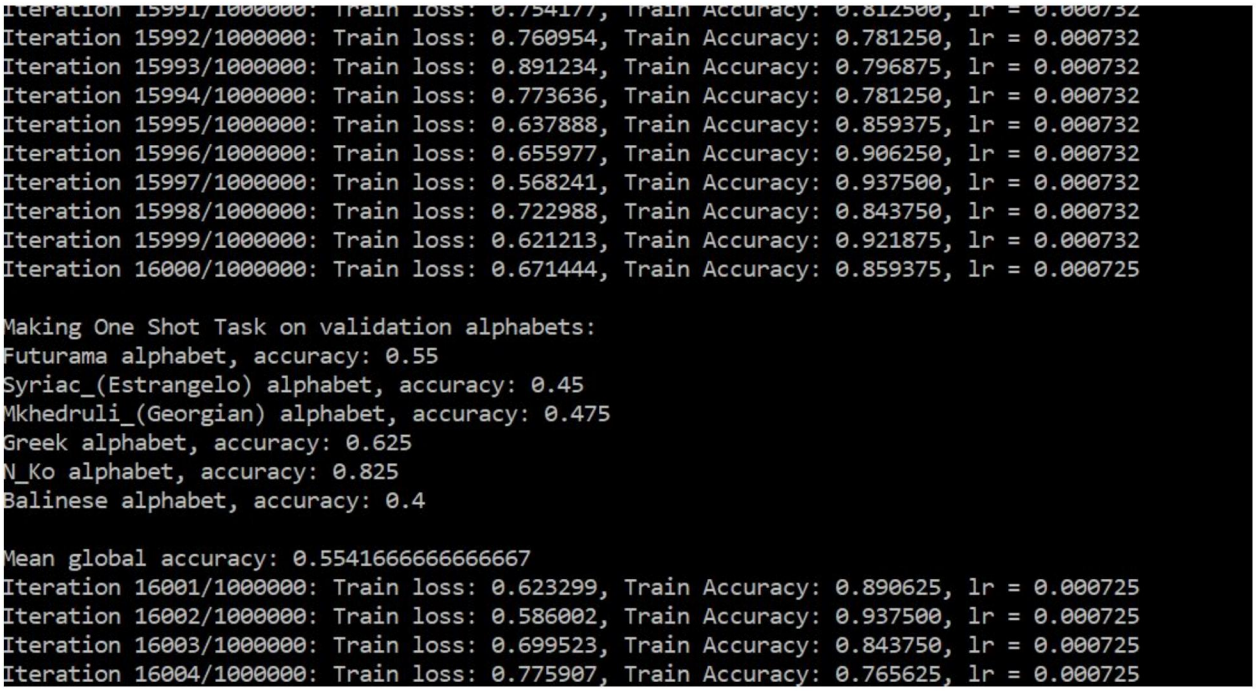
We have thus decided to use an alternate dataset - the AT&T “Database of Faces” found at “http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html”. There are ten different images of each of 40 distinct subjects.

Methods

Executing existing architecture on omniglot dataset

As a first step, we implemented the github’s implementation of the architecture on the omniglot dataset. From the 30 alphabets background set, 80% (24) are used for training and 20% (6) are

using for validation one-shot tasks. Since the network was taking very long to train, we only used trained it over 16,000 iterations. The results of this model are given below:



Execute the existing architecture on the AtT&T face dataset

We fully trained the existing architecture on the smaller AT&T dataset and observed the accuracy of results.

Approach:

The AT&T dataset has 40 different classes with 10 images each. These images are black and white pgm files. As a first step, we wrote a function to read pgm files into python. The images were of the size 112 X 92 so we padded them to convert them into 112 X 112 because the Koch et. al. architecture expects a square image.

We then split the data into training, validation and testing datasets in the 0.6,0.2,0.2 ratio and trained the model on the training dataset.

We tried different batch-sizes to increase the learning rate and finally used 100 as the batch size to train our model. After 2200 iterations, our training loss became constant at 0.75 after decreasing from 4.52. We stopped training the model at this stage and tested it's accuracy on a one-shot learning task on the test set images.

Results

We initialised our weights as random normal with mean of 0 and standard deviation of 1e-2 and biases as random normal with mean as 0.5 and standard deviation of 1e-2 (given in paper). We used the same architecture as the paper (shown in the picture above). Our primary metric was accuracy of correct detection in one shot and we got the following results:

Training set	95%
Validation set	65%
Testing set	66%

As we can see from above, the model is overfitting to the training data set and the accuracy on validation and testing data sets is not as good. We have used L2 regularisation but that has not prevented us from overfitting to the training dataset. One reason for this could be that we had very limited data and hence the chances of seeing the same data in further iterations was high and hence the chances of overfitting were high.

Conclusion

Siamese networks are really interesting networks and work really well for one shot learning. One shot learning has a lot of applications and if we had more time and computing resources we would have liked to train this model on a larger dataset and then tried to improve accuracy on the testing dataset by trying more regularization techniques like blocking.

Contribution

Ours is a two member team and both of us worked together on this project. Our contribution was equal.