Prediction of COVID-19 Cases using Siamese Network on Pair of X-Ray Images

Ahmed Dirir
Electrical Engineering and Computer Science
Khalifa University of Science and Technology
AbuDhabi, UAE
ahmed.m.dirir@gmail.com

***Abstract*—Machine learning algorithms outperformed many existing algorithms in many areas. In this research, the Siamese networks are utilized to build a solution that classify X-Ray images into positive cases and negative cases of COVID-19. The data sets contains few images most of which are positive samples. Therefore, the Siamese networks have been used on pairs of similar and dissimilar images. The results obtained give promising and outstanding performance. For the model optimization, Convolution Neural Networks are used to learn from the images the corresponding features that will help the Siamese network learn the distance function and similarity metric that will help separation of images into positive and negative samples.**

***Index Terms*—COVID-19, CNN, Siamese networks.**

I. Introduction

A Siamese neural network is a special type of neural network that shares the neural network weights across two input vector. During training, the network gets two inputs for each instance, forward propagate the two inputs through the network, calculate the difference between the two outputs, and update the weights using backpropagation. The network has only one weight matrix that is being shared across the two inputs. This architecture helps the neural network learn a distance function or a similarity between the two inputs. This helps a lot in cases where the number of training examples of some classes in the dataset is small. For instance, if the objective is to track a person in the street, the neural network will have few examples (images) of this person, therefore, the network will not be able to track the person. A better approach is to ask the network to learn the difference between the images of this particular person, and all other people. This will help since the neural network will learn a similarity metric or a distance function that will minimize the distance between instances of the same person, and maximize the distance between dissimilar instances.Many researches are going on currently trying to utilize the power of Siamese network especially when the data set contain fewer examples. Fig.1 shows a research tried to learn a distance function for evaluating similarity between graphs. Other research is trying to resolve the problem finding the proper and correct negative samples since most of the researchers before found it difficult to construct the negative samples [5].

Contrastive loss is a Distance-based Loss function as opposed to prediction error-based Loss functions like Logistic

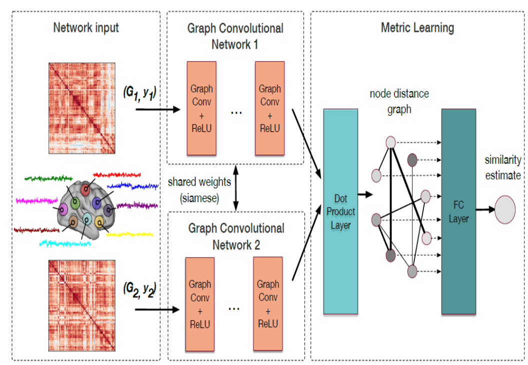

Fig. 1. Neural Network Model used to compare functional brain graphs[1].

loss or Hinge loss used in Classification. Similar to any distance-based loss, it minimizes the distance between semantically similar examples and finds an embedding that embeds them close together. The loss function has a term for similar instances distance minimization as well as a term for different instance distance maximization. Unlike conventional learning systems where the loss function is a sum over samples, the loss function here runs over pairs of samples. Let X_1, X_2 I be a pair of input vectors shown to the system. Let Y be a binary label assigned to this pair. $Y = 0$ if X_1 and X_2 are deemed similar, and $Y = 1$ if they are deemed dissimilar. Define the parameterized distance function to be learned D_W between X_1, X_2 as the Euclidean Distance between the outputs of G_W. That is,

$$D_W(X_1, X_2) = ||G_W(X_1) - G_W(X_2)||_2 \tag{1}$$

Then the general form of the loss function is

$$L = \sum_{k=i}^{n} L(Y, X_1, X_2)^i \tag{2}$$

$$L(Y, X_1, X_2)^i = (1 - Y)L_s(D_w)^i + Y L_D(D_w)^i \tag{3}$$

where (Y, X1, X2) i is the ith labeled sample pair, LS is the partial loss function for a pair of similar points, LD the partial loss function for a pair of dissimilar points. The final loss function is

$$L(Y, X_1, X_2) = (1 - Y)(D_w)^2 + Y * max(0, m - D_w^2) \tag{4}$$

Where m is a margin. The margin defines a radius around $G_W(X)$. Dissimilar pairs contribute to the loss function only if their distance is greater than this radius. The contrastive term involving dissimilar pairs, L_D, is important. Trying to minimize $D_W(X_1, X_2)$ on the set of all similar instances will probably lead to a non working solution, since D_W and the loss L could then be made zero by setting G_W to a constant. Most energy-based models require the use of contrastive term in the loss function. Fig.2 shows the margin plotted versus loss.

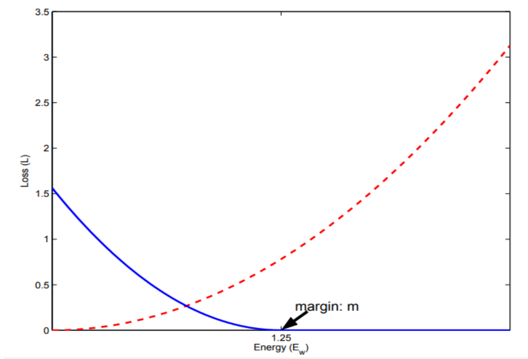

Fig. 2. Proper m margin learning vs loss.

II. Data Processing

The data set used in this project contains 11 different classes, most of which are COVID-19. The purpose of this project is to classify X-Ray images into positive and negative samples. Positive sample means that the patient will probably have COVID-19. As in the Fig.3, the positive samples contribute to more than 75% of the data set (280 positive sample, 70 negative samples). To resolve this, issue, only 70 images from the positive samples has been considered for training. However, the small number of images to train the neural network might results in under fitting since there are few examples for the network to learn. Siamese network comes in handy in this case, if we take a pair of X-Ray images as an input to the network, we will have 4900 negative samples, as well as 4900 positive samples. In total, the network will have 9600 examples for learning instead of 140 examples in the case of conventional neural network. Fig.4 shows negative samples of X-Ray images.

As mentioned before, the convolutional neural network that will be described in the next page will try to learn to representational space that will have the similar instances in between distance minimized. The representational space is like an embedding that transform the input vector to an embedding vector so that the distance between similar instances is minimum, as well as the distance between the dissimilar instances is maximum. Note that the second objective of Siamese network is that it allow the convolutional neural network to learn the features of the images. Therefore, two objectives have been satisfied. The distance function or the

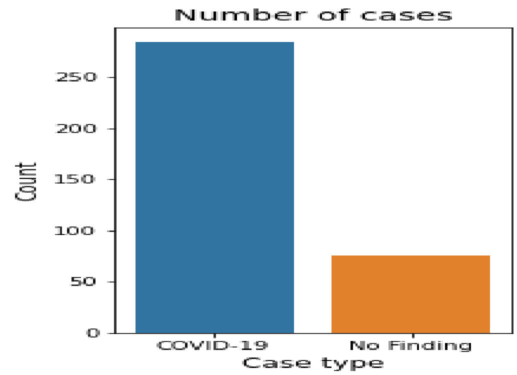

Fig. 3. Number of cases in each class.

similarity metric between the pairs of images is learned, as well as the features of the images is learned too. Afterward, the neural network when given an image to classify, it can use the learned feature, and distance function to find the distance between the given image, and all other images in the datasets that have known labels. The k-nearest neighbors is utilized to find the nearest neighbors, and based on the label of the nearest neighbors, the label of the given image can be known.

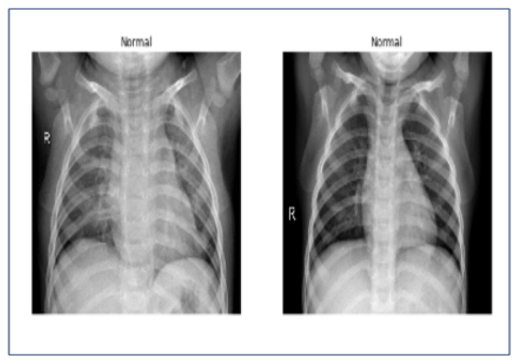

Fig. 4. Example of negative pair

III. Neural Network Model

The proposed neural network model is shown in the figure below, as explained, the model has two input layers of shape (150, 150, and 1). 150 is the size of the X-Ray image. Then each image is forwarded through the CNN model, and from the output, backpropagation will update the weights of the CNN model. Note again that the weight matrix W is the same for the two inputs; the network is same for first input as well as the second input. Since the output is zero for the similar pairs, and the output is one for dissimilar pairs, the CNN model will extract the proper features from the images that model can classify based on. If the weights are not shared, then, there is no point of the whole process.

The CNN model shown in the left side of this page is a conventional CNN, which has BacthNorm that is helpful when the CNN model is deep. MaxPooling is used too in

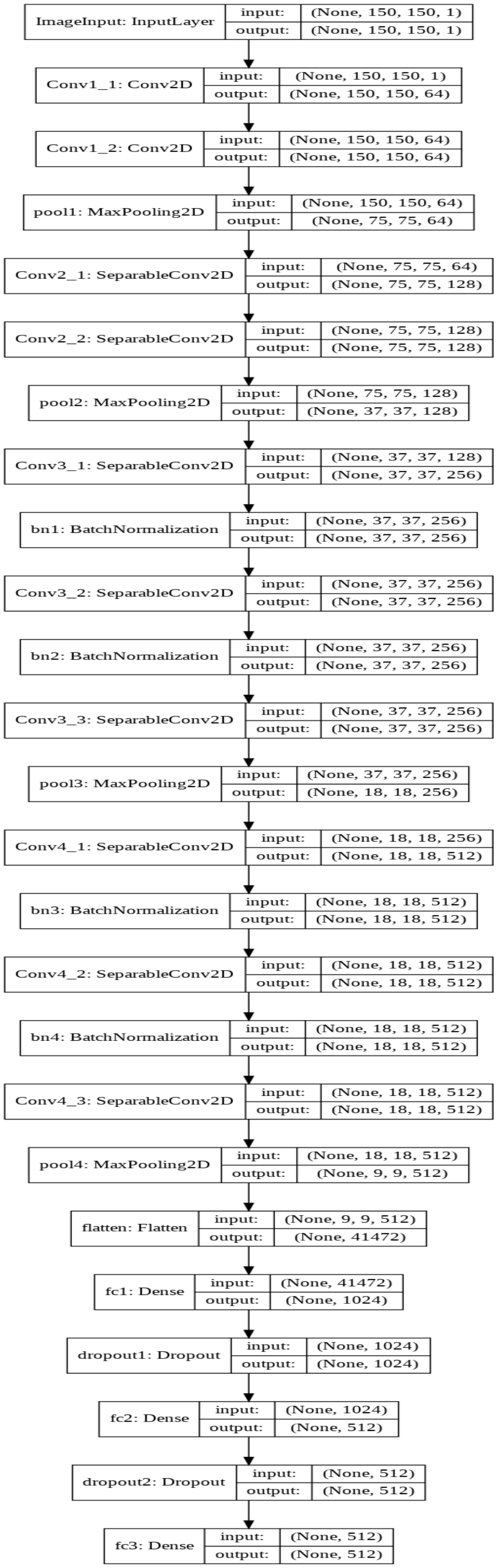

Fig. 5. Convolutions Neural Network Model.

CNN to help build an abstract representation of the input that reduce the dimensionality of the input too. SeparableConvNet [3] is an alternative for the conventional ConvNet, but it introduces fewer parameters. This will help in reduction of the computation cost as well as increase the learning speed. This CNN architecture is based on [4].

IV. Results Discussion

The training and validation losses are shown in the figure below. The validation loss starts at 0.55, and slowly decreases through the different epochs. The plot is showing the results for the first 100 epochs. The validation loss at the end of the training is around 0.2. Although the curve has many zigzag, and overshooting, the net validation loss is going down as the number of epoch's increases. The main issue faced in the project is the limited GPU computing resources given by the google Colab.

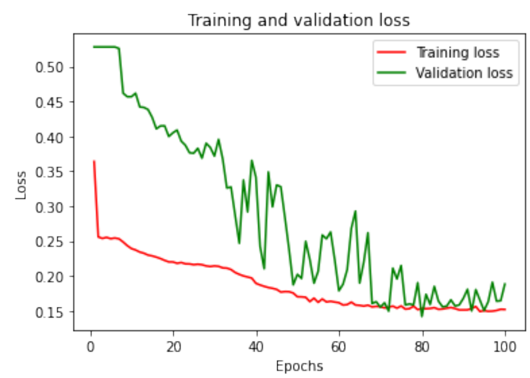

Fig. 6. Training and validation Loss.

Similarly, the validation accuracy is increasing as the number of epoch's increases. Regardless of the overshoots and undershoots, the net validation accuracy is increasing. The final validation accuracy is 70

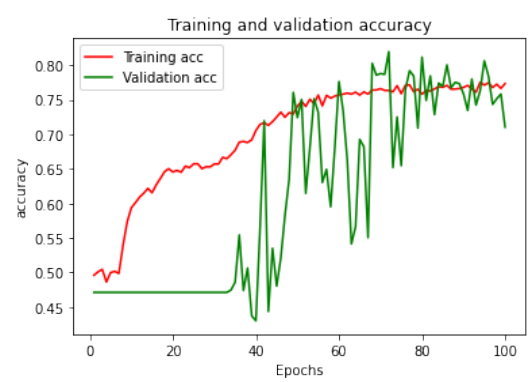

Fig. 7. Training and validation Accuracy.

The testing accuracy in the other hand was a bit lower than the validation accuracy. Fig.8 is a plot for the area under the curve for the testing pairs. The testing accuracy is 66%. The main reason for the testing accuracy not reaching higher value is the lower number of epochs. If the number of epochs is increased as well as scheduling of the learning rate is carried on, the testing accuracy for sure will be high.If the number of epochs is increased as well as scheduling of the learning rate is carried on, the testing accuracy for sure will be high.

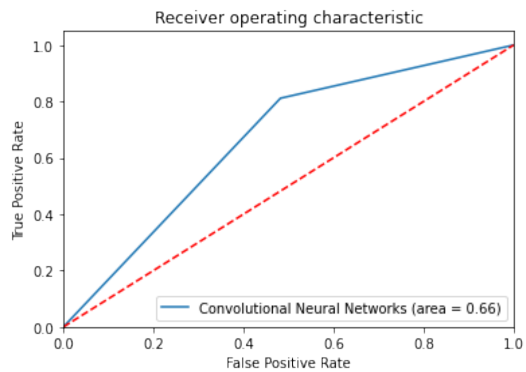

Fig. 8. Area Under Curve for testing examples.

Finally, the confusion matrix for the testing pairs is shown in the figure below. The confusion matrix shows that the network is good at predicting positive samples (80% accuracy), but not good at predicting negative samples (50%). Maybe because of the way the negative pairs has been generated. Since the negative pairs is the collection of the mutual pairs from the same class which might create some unwanted variance in the negative sample.

Fig. 9. Confusion matrix for testing examples.

Conclusion

Finally, this research tried to use the proposed Siamese network to find the cases that have COVID-19. The data set contained few examples of positive and negative classes. The results obtained shows the strength of Siamese network in learning a distance function as well as extracting features from the example.

References

[1] Sofia Ira Ktena, Sarah Parisot, Enzo Ferrante, Martin Rajchl, “Distance Metric Learning using Graph Convolutional Networks: Application to Functional Brain Networks” (MICCAI) 2017

[2] Raia Hadsell, Sumit Chopra, Yann LeCun “Dimensionality Reduction by Learning an Invariant Mapping” CVPR 2006

[3] Franc¸ois Chollet “Xception: Deep Learning with Depthwise Separable Convolutions” Google 2017

[4] Beating everything with Depthwise Convolution”, Kaggle.com, 2020. [Online]. Available: https://www.kaggle.com/aakashnain/beating-everything-with-depthwise-convolution. [Accessed: 14- May- 2020].

[5] Xiaoke Zhu, Xiao-Yuan Jing, Fan Zhang “Distance learning by mining hard and easy negative samples for person re-identification” Elsevier 2019.