Gurjus Singh

MSDS 458 – Artificial Intelligence and Deep Learning

March 14th, 2021

Week 10: A.4 Fourth Research/Programming Assignment

**Abstract**

In this research, I came up with a way to classify X-Ray Chest Images with COVID-19. The goal of this research was to use supervised learning binary classification with Normal and COVID-19 X-Ray Images. I used a CNN architecture as this was important in image classification. I used accuracy for deciding which model was performing the best, but also looked at other metrics in my best model to see how it was performing. I also performed T-SNE on my best model as well as saw how the Convolution/Max Pooling by making a channel grid.

**Introduction**

In purpose of this research was to use supervised learning binary classification to classify X-Ray Chest Images of suspected COVID-19 patients. The dataset I used was from Kaggle and contained 2000 train images and 328 test images divided evenly between COVID Positive and COVID Negative images. The Neural Network architecture I used was a Convolution Neural Network architecture as this was suitable for image data. I tried out different CNN architectures before choosing my best model. I also decided to find out what features the layers of the CNN were learning by creating a grid of channels. I also decided to perform T-SNE on my best model. After experimenting with my best model, I then made a management recommendation.

**Literature Review**

As COVID-19 spreads, and vaccines are distributed, there continues to be growing research on how to diagnose patients. One such paper that I found, talked about how countries have adopted reverse transcription polymerase chain reaction to diagnose COVID-19 [1]. The authors of the paper point out this test takes 4-6 hours or even a whole day to get results [1]. The test can also give false positives and false negatives [1]. The authors then point out that one solution to preventing false positives/negatives is to test COVID-19 infections by using XRAY/CT Scans of the Chest in COVID patients [1].

The paper goes on to share the results of XRAY/CT Scan classification [1]. They use multi-image classification to decrease overfitting [1]. This is a way to generate more images for the CNN architecture to train on [1]. In the results of the experiments with a 70 percent-30 percent training-testing split the authors obtained a 95.38 percent accuracy for CT Scans, while for X-ray they obtained a 98.97 percent accuracy [1].

**Methods**

I first start out this research by importing the necessary packages to complete this project. I found the keras and tensoflow packages important for implementing the CNN architecture. I also found sklearn package important for computing metrics. I also found matplotlib package important for creating the EDA, and the os package was important for extracting the files from the directories. The imported packages are shown below in 1-1.

Text, application

Description automatically generated

*Import Packages 1-1*

After importing the packages my next step was to do some EDA. I plotted the first 5 images of COVID-POSITIVE patients as seen in 1-2 and plotted first 5 photos of COVID NEGATIVE patients as seen in 1-3.

Calendar

Description automatically generated

*COVID-POSITIVE PATIENT XRAYS 1-2*

*Graphical user interface, application

Description automatically generated*

*COVID-NEGATIVE PATIENT XRAYS 1-2*

After taking an initial look at the data, the next thing I did is loaded the data in the RGB format and also created labels which we will use for training. I also resized each image to 64 x64. After loading of the images and creation of the labels, I looked how many COVID-POSTIVE and NEGATIVE X-RAY Images were in the dataset as seen in 1-3. I noticed there were even number of both sets of images.

Chart, bar chart

Description automatically generated

*Number of COVID CASES 1-3*

I also wanted to see what the Train and Test split percentage was as it was already split when taking the dataset from Kaggle. What I noticed was that it was a 85.9 percent to 14.1 percent Train to Test split as seen in 1-4.

Chart, pie chart

Description automatically generated

*Training-Test Split 1-4*

After looking at the split I then extracted the labels I created and encoded and stored them in an array, I did the same with the images. I then normalized the images by dividing each image by 255 to get it between 0 and 1. After normalizing the images, I then added more images to the training data to prevent overfitting by using an ImageDataGenerator. This is known as Data Augmentation, and it creates more images by manipulating the existing images to create more images for training. After doing data augmentation, I then split the training data 80 percent to 20 percent to create a validation set.

I then went ahead to create my CNN Architectures which involved Max Pooling, Convolution and Average Pooling layers. Max pooling helps in reducing the number of parameters in the network by “downsizing feature maps” by only extracting the important features of an image[2]. The convolutional layer functions in that tries to extract local patterns from a window [2]. The window is called a filter which goes over each part of the image extracting the features in the image [2]. Average pooling retains less important information of an image by averaging the features [4].

After building each CNN Architecture, I then observed the accuracy and obtained the best model. On the best model, I performed TSNE, also built a grid for each layer in the CNN architecture, and computed other metrics such as F1 Score, Precision Score etc.

**Results**

In the research I performed a total of **17 experiments, 20 Epochs each**, and also performed regularization techniques such as L1, L2 Regularizers, Dropout and Early Stopping. My Performance Summary is below in 1-5. What I noticed from these experiments was that **Experiment 9 with 3 Max Pooling, 3 Convolution Layers with 128 Filters; and 1 Dense Layer with 256 Neurons** was performing the best. I noticed it’s testing accuracy was 98.47 while Training accuracy was 98.50 which means it was not overfitting. This was the highest testing accuracy that I had among the 17 models which was why I chose this model.

Table

Description automatically generated with medium confidence

*Performance Summary; See Excel Sheet attached for better look 1-5*

Next, I looked into the metrics to see how good of a job the model 9 was doing. From the results in 1-6, 1-7. From the Confusion Matrices, I felt like the model was doing a fantastic job. I also noticed that Precision and Recall scores were around 0.98, and the F1 Scores were high as well being close to 1 which is the best score.

Graphical user interface, application

Description automatically generated

*Training Set Metrics 1-6*

*Graphical user interface, application

Description automatically generated*

*Testing Set Metrics 1-7*

After looking at key metrics for my model, I then looked at the grid for how the layers in the architecture learned in 1-8. I noticed the layers was trying to learn the features of the bones in the X-ray.

A picture containing text, sign

Description automatically generated

*Grid for Layers 1-8*

Lastly, I performed T-SNE, which is dimensionality reduction unsupervised learning technique to visualize higher dimension data as seen in 1-9 [3].

Chart, scatter chart

Description automatically generated

*T-SNE on COVID Dataset 1-9*

**Conclusion**

In this experiment, I did binary classification on COVID-19 Xray images using CNN Architecture. From doing 17 experiments I chose Experiment 9 which had an architecture of **3 Max Pooling, 3 Convolution Layers with 128 Filters; and 1 Dense Layer with 256 Neurons** as the best experiment. After choosing the best model, I also computed key metrics in 1-6, 1-7, saw how the layers learned by creating a grid in 1-8, and performed T-SNE in 1-9**. For my recommendation to management,** I choose an architecture of **3 Max Pooling, 3 Convolution Layers with 128 Filters; and 1 Dense Layer with 256 Neurons** to diagnose COVID Patients. In the future, I hope to see if the CNN Architecture can distinguish between other fluid like illnesses like Pneumonia.

References

[1] Purohit, K., Kesarwani, A., Kisku, D. R., & Dalui, M. (2020, January 1). *COVID-19 detection on chest x-ray and ct scan images using multi-image augmented deep learning model*. BioRxiv. <https://www.biorxiv.org/content/10.1101/2020.07.15.205567v2.full>

*[2]* Chollet, F. (2017). *Deep Learning with Python*. Manning Publications Company.

*[3] Introduction to t-SNE*. (n.d.). Data Camp. Retrieved February 6, 2021, from <https://www.datacamp.com/community/tutorials/introduction-t-sne>

*[4]* C. (2020, January 30). *What are Max Pooling, Average Pooling, Global Max Pooling and Global Average Pooling? –*. MachineCurve. https://www.machinecurve.com/index.php/2020/01/30/what-are-max-pooling-average-pooling-global-max-pooling-and-global-average-pooling/