CovidCT image classification

Yuchen JIANG

Janurary, 2021

1 Introduction

Project Proposal

Coronavirus disease 2019 (COVID-19) has infected more than 100 million individuals all over the world and caused more than 2 million deaths. One major hurdle in controlling the spreading of this disease is the inefficiency and shortage of medical tests. To mitigate the inefficiency and shortage of existing tests for COVID-19, we combine GAN and transfer learning methods to diagnose COVID-19 based on CT images.

DataSet

The whole COVID dataset has two classes: positive and negative. And the task is to classify each CT image into one class, which is a binary classification problem. The original dataset is split into three sets, including training set, validation sat and test set. And there are 425 images in the training set, 118 in the validation set and 203 in the test set.

Work Overview

In Chapter 2, we will focus on GAN model, including the particular networks architecture, the detail of the training process.
In Chapter 3, we will state a pre-trained model named visual transformer(ViT), describe the experiments in detail and the obtained results.
Chapter 4 contains the conclusion of our work.

2 GAN Model

Deal with the small datasets is a recurrent problem in the medical imaging domain, especially when employing supervised machine learning algorithms that require labeled data and larger training examples. Although public medical datasets are available online, most datasets are still limited in size and only applicable to specific medical problems, the reason is that collecting medical data is a complex and expensive procedure and researchers use to attempt to overcome this challenge by using data augmentation. The most classical data augmentation methods include simple modifications of dataset images such as translation, rotation, flip and scale, which is a standard procedure in computer vision tasks. However, this is useless because little additional information can be gained from small modifications to the images.

Synthetic data augmentation of high quality examples is new, sophisticated type of data augmentation. Synthetic data examples learned using a generative model enable more variability and enrich the dataset to further improve the system training process. One such promising approach inspired by game theory for training a model that syntheses images is known as Generative Adversarial Networks (GAN).

2.1 Architecture

We use Deep Convolutional GAN (DCGAN) where both the Generator (G) and Discriminator (D) networks are deep CNNs. The model consists of two neural networks that are trained simultaneously. The first network is termed the discriminator. The role of the discriminator is to discriminate between the real and fake samples. It is inputted a set of sample x_1,..., x_m and outputs $D(x_i)$, $\forall i \in [1, ..., m]$, its probability of being a real sample. The second network is termed the generator. The generator synthesizes samples that D will consider to be real samples with high probability. G gets input samples z_1,..., z_m from a known simple distribution p_z, and maps G(z) to the image space of distribution p_g. The goal of G is to achieve $p_g = p_{data}$.
Adversarial networks are trained by optimizing the following loss function of a two-player minimax game:

$$\min_G \max_D \mathbb{E}_{x \sim p_{data}} log D(x) + \mathbb{E}_{x \sim p_x}[log(1 - D(G(z)))]$$

The discriminator is trained to maximize D(x) for images with $x \sim p_{data}$ and to minimize D(x) for images without $x \sim p_{data}$. The generator produces images G(z) to fool D during training such that $D(G(z)) \sim p_{data}$. Therefore, the generator is trained to maximize D(G(z)), or equivalently minimize 1 - D(G(z)). During training the generator improves in its ability to synthesize more realistic images while the discriminator improves in its ability to distinguish the real from the synthesized images. Hence the moniker of adversarial training.

2.2 Training Process

batch size: Considering the size of the dataset we chose to train our model on we took a proper batch size of 32.
transforms of training set: we resize the input images to the desired image(64×64) size before providing to the network, then crop the given image at the center, and normalize the given image to have unit mean as well as unit standard deviation.
hyperparameters: We apply stochastic gradient descent with the Adam optimizer, an adaptive moment estimation that incorporates the first and second moments of the gradients, controlled by parameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$ respectively. Learning rate of 0.0002 is kept for the discriminator network and 0.002 for the generator one. And we decided to save, record and consider during experiments the results of the process 200 epochs.

3 ViT Model

Self-attention-based architectures, in particular Transformers is more and more popular in deep learning field, and multiple works try combining CNN-like architectures with self-attention and some replacing the convolutions entirely in computer vision. In this case, we employ a standard transformer model called ViT[1] to solve the CovidCT image classification.

[1]This model can be retrieved from: https://arxiv.org/pdf/2010.11929.pdf

3.1 Architecture

In this model, we split an image into patches and put the sequence of linear embeddings of patches into a transformer like what has been done in NLP.
We reshape an image $\mathbf{x} \in \mathbb{R}^{H\times W\times C}$ into a sequence of patches $\mathbf{x}_p \in \mathbb{R}^{N\times(P^2C)}$, where (H, W) is the resolution of the original image, C is the number of channels, (P, P) is the resolution of each image patch, and $N = HW/P^2$ is the resulting number of patches, and then we flatten the patches and map to constant dimensions with a trainable linear layer. Then we get an output as the image representation after the transdormer encoder. All other steps are like BERT classification tasks. And the whole processing is in Figure 1.

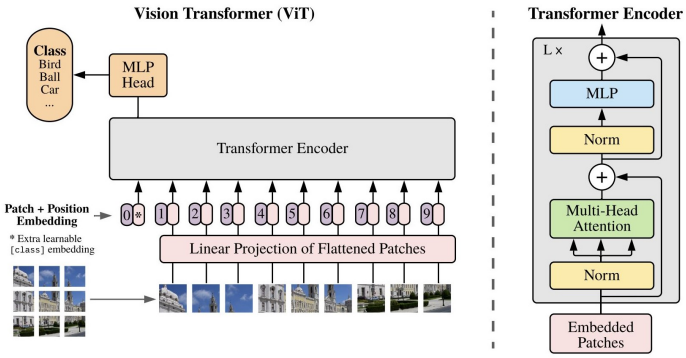

Figure 1: Vision Transformer Model[2]

3.2 Training Process

Considering the computing power and limited time, We use ViT-B/16 pre-trained on imagenet21k dataset.
batch size: At this time, we took a proper batch size of 16.
transforms of training set: we resize the input images to the desired image(224×224) size before providing to the network, then crop the given image at a random location. Because we convert three dataset into colorful ones, so we let the brightness and contrast of each image equal to 0.2, finally normalize the given image to have unit mean as well as unit standard deviation.
hyperparameters: We apply adam optimizer, learning rate of 0.0001 is kept for the network. And we decided to save, record and consider during experiments the results of the process 20 epochs in small dataset and 30 in large dataset.
valuation: We use F1-score to valuate the model, which is computed by the harmonic mean of precision and recall, where precision is the fraction of true positives among the predicted positives and recall is the fraction of the total number of true positives that are predicted as positive.
number of generated images: We use F1 score to detect the influence of different numbers of generated figures in two train loaders(Covid and NonCovid) on the experimental results and we choose 2000 in among these experiments. The results are shown below:

Generated Number	500	1000	2000	2500	3000
F1 Score	0.8035	0.8259	0.8370	0.75372	0.8018

[2]This figure is retrieved from: https://arxiv.org/pdf/2010.11929.pdf

4 Conclusion

In this project, a GAN learning for COVID-19 detection in limited CT images is presented. The lack of benchmark datasets for COVID-19 CT images was the main motivation of this project. The main idea is to use the GAN network to generate more images to help in the detection of the virus from the available CT images. The task of binary classification and detection is quite challenging in the absence of a large dataset. It was shown how the generated synthetic images could be used to augment the original dataset and eventually lead to a higher classification and detection accuracy of the CT images.
Then we try a recently popular model related to transformer architecture in binary classification task to solve running time without reducing F1 score. The main point is to reshape each image into a sequence of patches like tokens in BERT model.
We would like to further investigate a small range of generated figures and specific numbers of epoch into our experiments. Also how to add transformer into GAN model could be a priority to investigate in future.