CS542 Class Challenge Report

Zhihui Zhang, Shicong Wang

I. Introduction

In this class challenge, we classify X-ray images.The data we use has been collected by Adrian Xu, combining the Kaggle Chest X-ray dataset with the COVID-19 Chest X-ray dataset collected by Dr. Joseph Paul Cohen of the University of Montreal.

For Task 1, we train a deep neural network model to classify normal vs. COVID-19 X-rays using the data we collect above. After That, we visualize features of training data by reducing their dimensionality to 2 using t-SNE.

For Task 2, we train deep neural network models to classify an X-ray image into one of the following classes: normal, COVID-19, Pneumonia-Bacterial and Pneumonia-Viral and make comparisons among multiple model architectures. After training is complete, we visualize features of training data by reducing their dimensionality to 2 using t-SNE.

II. Task 1

Test loss**: 0.0314907506108284**
Test accuracy**: 1.0**

Deep Neural Network Architecture

For binary classification, we started with training a VGG16 architecture using pretrained weights from "Imagenet". VGG19 architecture is chosen over other models as it has skipped connections between layers that add the output from previous layers to the output of stacked layers which helps us train deeper networks.

We then add a flatten layer to feed them as inputs to the dense layer, and then add a Dense layer with 404 neurons and ReLu activation to introduce non-linearity in the model. We next add a dropout layer with a rate of 0.25 to avoid overfitting. After that, we continuously add a Dense layer with 66 neurons and ReLu activation, followed by another dropout layer with a rate of 0.25. We subsequently add a Dense layer with 66 neurons and ReLu activation. Lastly add another dense layer with one neuron and the sigmoid activation function as we are conducting a binary classification problem.

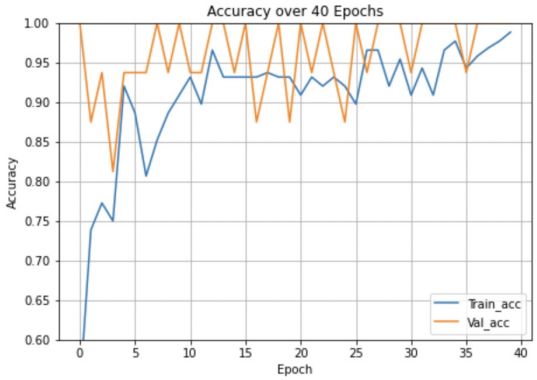
```
Model: "sequential_8"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg16 (Functional)          (None, 7, 7, 512)         14714688

 flatten_7 (Flatten)         (None, 25088)             0

 dense_20 (Dense)            (None, 404)               10135956

 dropout_8 (Dropout)         (None, 404)               0

 dense_21 (Dense)            (None, 66)                26730

 dropout_9 (Dropout)         (None, 66)                0

 dense_feature (Dense)       (None, 66)                4422

 dense_22 (Dense)            (None, 1)                 67

=================================================================
Total params: 24,881,863
Trainable params: 10,167,175
Non-trainable params: 14,714,688
_________________________________________________________________
```

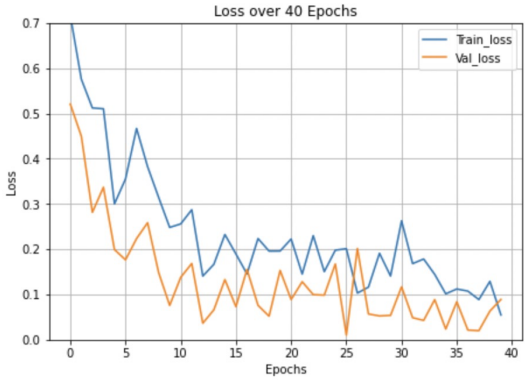
Optimizer, Loss Function, Parameters and Regularization

The optimizer used in this model is the Adam algorithm with a learning rate of 0.0002. Binary Crossentropy loss function is applied since we are attempting to conduct a binary classification between Normal and Covid-19 Chest X-Rays. Drop out serves as the regularization technique in the model, which aims at preventing overfitting.

Accuracy and Loss

From the accuracy plot, we find that the validation accuracy maintains a high level after some previous epochs. From the loss plot, we can see that with each new iteration, validation loss is approximately equal to training loss which shows that the model is converging in very few epochs. We can conclude from two plots that the model is not overfitting.





We calculate the confusion matrix. And here is the result.

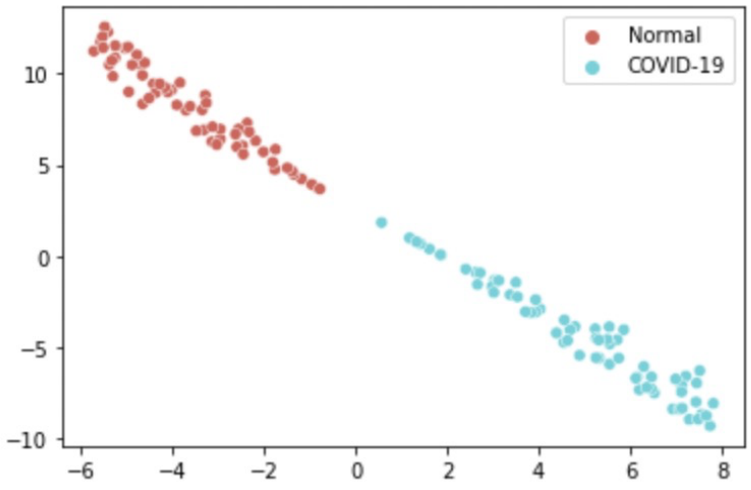
Confusion Matrix

	precision	recall	f1-score	support
Covid	1.00	1.00	1.00	9
Normal	1.00	1.00	1.00	9
accuracy			1.00	18
macro avg	1.00	1.00	1.00	18
weighted avg	1.00	1.00	1.00	18

The predictions are the same as the ground truth labels.

t-SNE Visualizations

As shown in the plot, it's apparent to see the t-SNE 2-D visualization for two classes, Normal vs. Covid-19 Chest X-rays. The extracted features are good since the data points representing the two classes appear in the two distinct clusters.



III. Task 2

Test loss**: 0.6563900709152222**
Test accuracy**: 0.89**

Deep Neural Network Architecture

In this task we trained a deep neural network to classify an X-ray image into one of the following classes: normal, COVID Pneumonia-19, Pneumonia - Bacterial and Pneumonia - Viral using the data mentioned above.
The model with the highest accuracy is Densenet201 with pre-trained weights from 'imagenet'. We also add a dense layer with 254 neurons and ReLU activation followed by a dense layer with 64 neurons and ReLU activation. There is a dropout layer with a dropout rate of 0.2 to avoid overfitting before the output layer. The output layer consists of 4 neurons and the softmax activation function which is designed for the multi-class classification.

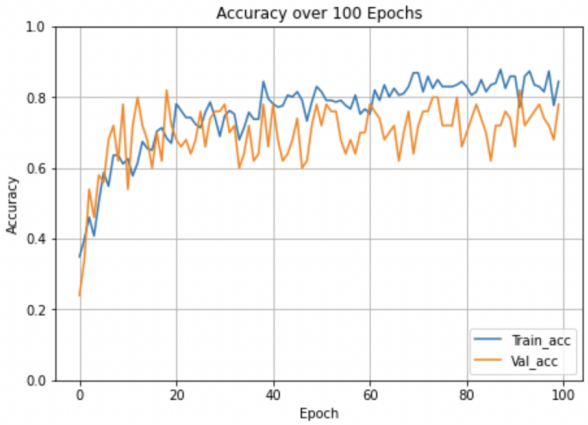
```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 densenet201 (Functional)    (None, 7, 7, 1920)        18321984

 flatten_1 (Flatten)         (None, 94080)             0

 dense_2 (Dense)             (None, 254)               23896574

 feature_space (Dense)       (None, 64)                16320

 dropout_1 (Dropout)         (None, 64)                0

 dense_3 (Dense)             (None, 4)                 260

=================================================================
Total params: 42,235,138
Trainable params: 23,913,154
Non-trainable params: 18,321,984
_________________________________________________________________
```

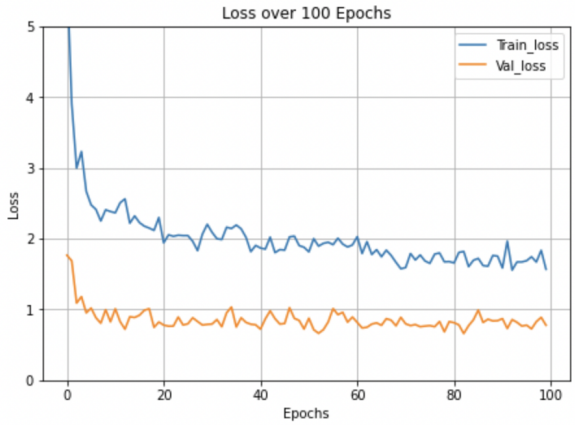
Optimizer, Loss Function, Parameters and Regularization

The optimizer used in this model is the Adam algorithm with a learning rate of 0.0002 as it is computationally efficient. We use the categorical cross entropy as our loss function with a label smoothing rate of 0.09. We use the Dropout technique to prevent overfitting. We also assign the class weights to be {0:1.0, 1:1.0, 2:3.0, 3:5.0} in order to address the lower accuracy of Pneumonia - Bacterial and Pneumonia - Viral class.

Accuracy and Loss

In the accuracy plot for the model we can see that the validation accuracy is higher than the training accuracy. In the loss plot for the model we can see that with each new iterating validation loss is lower than training loss which shows that the model is converging in very few epochs. We can conclude from both accuracy and loss plots that the model is not overfitting.





We calculate the confusion matrix. And here is the result.

	precision	recall	f1-score	support
covid	1.00	1.00	1.00	9
normal	1.00	0.89	0.94	9
pneumonia_bac	0.86	0.67	0.75	9
pneumonia_vir	0.75	1.00	0.86	9
accuracy			0.89	36
macro avg	0.90	0.89	0.89	36
weighted avg	0.90	0.89	0.89	36

Comparison among different models

In this section, we will illustrate our model training and hyperparameter tuning process.

1. Try different pre-trained models
   Since the size of the dataset is small, it is hard to train a deep convolutional neural network with high accuracy. Therefore, we will try multiple pre-trained models without adding additional fully connected layers between the flatten layer and the output layer. From the table below, we will pick ResNet152V2 and DenseNet201 as our baseline models for future training.

Model	Accuracy
DenseNet201	0.722
InceptionV3	0.638
ResNet152V2	0.694
VGG16	0.611
VGG19	0.611

2. Layers Design
   Starting from a layer with 2^n neurons, I tried multiple combinations of the number of neurons in each layer. Based on some previous papers, we will start with adding two layers with the same number of neurons and then decreasing the number of neurons in the latter layer. As we do not have too much data in our training sets, we think 256 neurons in each layer will be a good start. We will continue using the Densest201 model with two fully connected layers with 254 and 64 neurons as well as ResNet152V2 model with two fully connected layers with 256 neurons in the later hyperparameter tuning process.

Model	Layer structures	Accuracy
DenseNet201	FC1: 256 + FC2: 256	0.750
	FC1: 256 + FC2: 128	0.778
	FC1: 256 + FC2: 64	0.806
	FC1: 254 + FC2: 64	0.833
ResNet152V2	FC1: 256 + FC2: 256	0.750
	FC1: 256 + FC2: 128	0.694
	FC1: 256 + FC2: 64	0.667
	FC1: 254 + FC2: 64	0.667

3. Dropout
   Adding a dropout layer after a fully connected layer can help prevent overfitting. The accuracy of ResNet152V2 decreases dramatically after adding a dropout layer after two fully connected layers. I will choose the models with highest accuracy after applying dropout as my later tuning process.

Model	Layer structures	Drop out rate	Accuracy

DenseNet201	FC1: 254 + FC2: 64	0.2	0.722
		0.5	0.694
ResNet152V2	FC1: 256 + FC2: 256	0.2	0.5
		0.5	0.611

4. Learning rate
   The learning rate is a hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are updated. We tried different learning rates to see how well our models are.

Model	Learning rate	Accuracy
DenseNet201	0.0002	0.833
	0.001	0.777
ResNet152V2	0.0002	0.556
	0.001	0.639

5. Class weights & label smoothing
   Noticed that our models performed poorly when classifying the Pneumonia cases. As our final model, I assigned more weights for Pneumonia classes to force the model to put more emphasis on classifying Pneumonia - Bacterial and Pneumonia - Viral. We first add a label smoothing rate of 0.09 to the loss function and we multiple combinations of class weights.

Model	Class weights	Accuracy
DenseNet201	{0: 1.0, 1: 1.0, 2: 3.0, 3: 3.0}	0.720
	{0: 1.0, 1: 1.0, 2: 3.0, 3: 5.0}	0.889

t-SNE Visualizations

As we can see in the graph below the t-SNE 2-D Visualization for the four classes: Normal, COVID-19, Pneumonia-Bacterial and Pneumonia-Viral Chest X-rays. The extracted features are good as there is a clear distinction between the Normal, COVID-19 and Pneumonia classes. However, the model isn't able to distinctly classify Pneumonia-Bacterial and Pneumonia-Viral as it is not able to recognize the specific parameters that distinguish the two classes.