PNEUMONIA

Definition

An infection in one or both lungs, caused by bacteria, virus or fungi. It involves inflammation of alveoli and causes difficulty in breathing. With Pneumonia, air sacs will become fluid or pus. The disease can be life threatening, particularly to infants and people above 65 years of age.

Symptoms

Symptoms can be,

1. Cough
2. Fever, sweating and chills
3. Shallow breathing
4. Chest pain

Cause

Causes for Pneumonia are,

1. Bacteria - **Streptococcus pneumoniae**
2. Influenza
3. SARS-CoV2 (virus that cause Covid-19)

Risk Factors

1. Conditions that weaken the immune system. Eg. AIDS
2. **Chronic Obstructive Pulmonary Disease** (COPD)
3. Diabetes
4. Old Age
5. Lack of proper Nutrition

How to determine via Chest X-rays

When interpreting the x-ray, the radiologist will look for white spots in the lungs (called infiltrates) that identify pneumonia.

DATASET

Structure

The dataset is organized into three folders - train, val and test, each having two subfolders (for corresponding class) **NORMAL** and **PNEUMONIA**.

Source

The chest X-ray images were taken from pediatric patients from **Guangzhou Women and Children's Medical Center, China**. All chest radiographs were initially scanned for quality control by removing the low quality or unreadable scans.

Link for the dataset - Kaggle - Chest X-ray Images

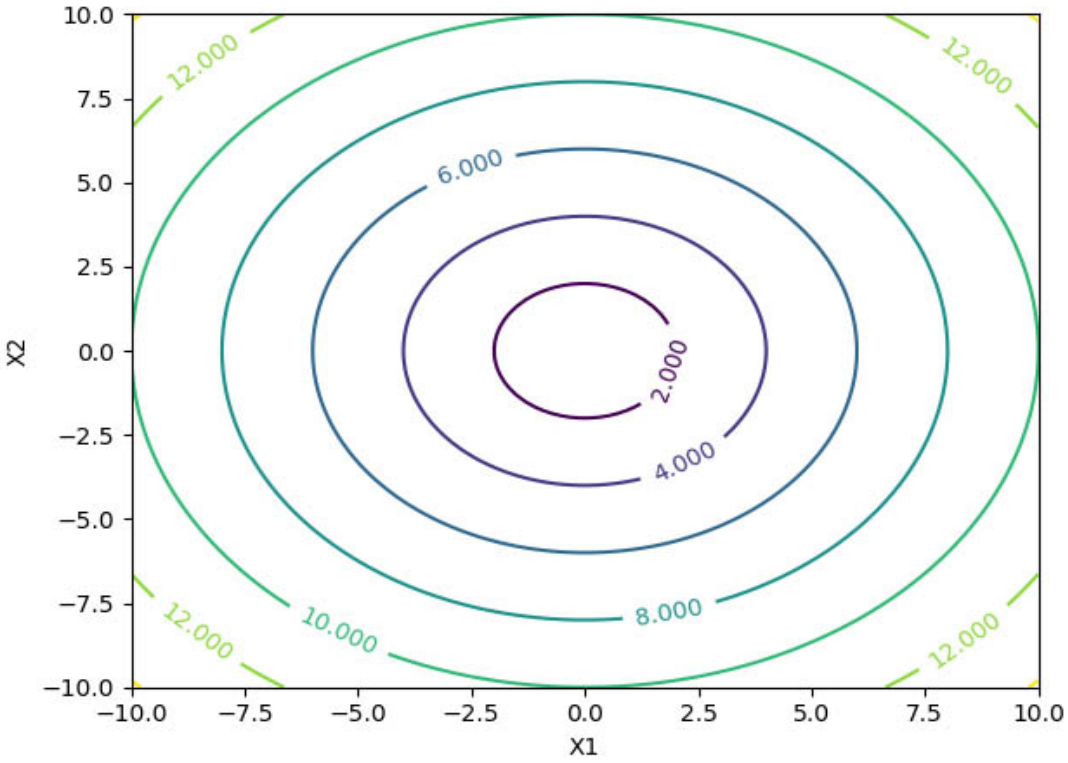
LAYERS

Convolution layer

Kernels or filters are applied on this layer, which are convolved with portions of images to capture specific features (some capture edges, others capture contrast, etc). This also assists in reducing image size. Filters can be considered as weights for images (which they are sort of).

BarchNormalization

It happens that the cost function curve may thin out at some axis. So this normalizes the Inputs for faster training. This causes the cost curve to look something like the figure below.



DropOut Layer

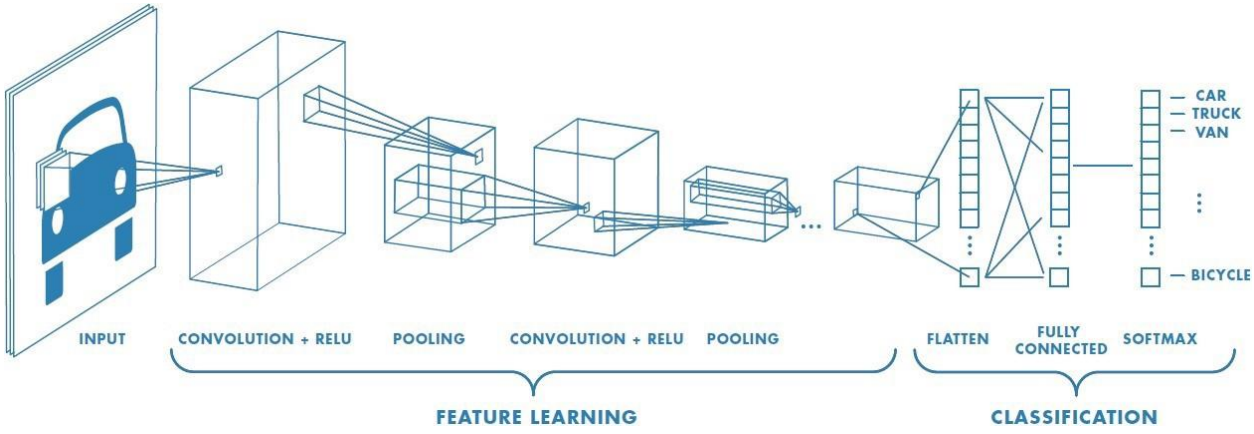
Assists in regularizing the weights by randomly deleting some units, so that the other nodes don't get too accustomed to the inputs and are able to take variations into account.

Max Pooling Layer

Move a window along the image to capture the dominant features in the image. Here the window size is (2,2).

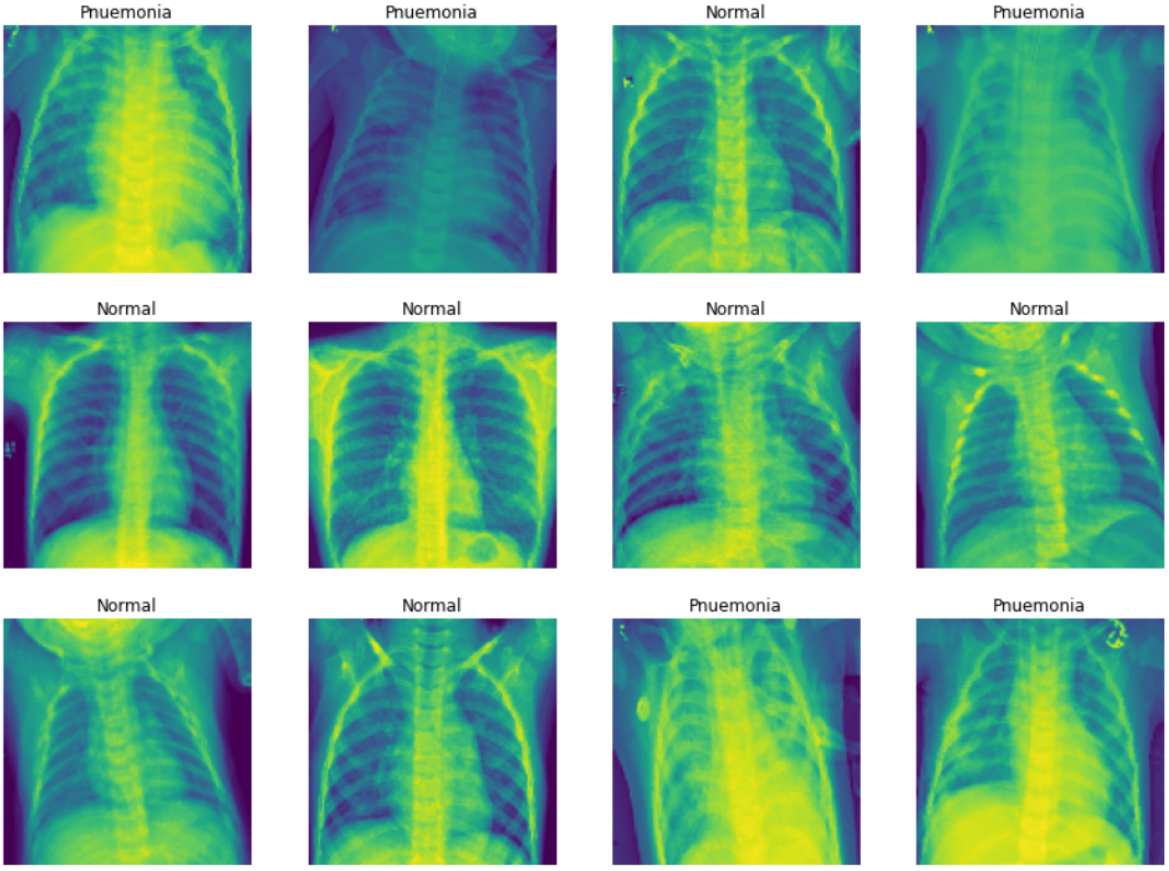
How does CNN work ?

A simple neural network when used for image classification may end up with a large amount of feature set. Convolutional Neural Networks, on the other hand, involves convolving the image with a kernel. The kernel can be considered as the weights for the image and they get updated while backpropagating. Another important layer used in CNN is the pooling layer (maxpooling or averagepooling) which only focuses on the dominant trait in the image and so filters out other information. This results in model learning only the dominant traits of the image, while reducing the size of the data. After this the new image can be flattened and fed to the Dense layer for prediction.

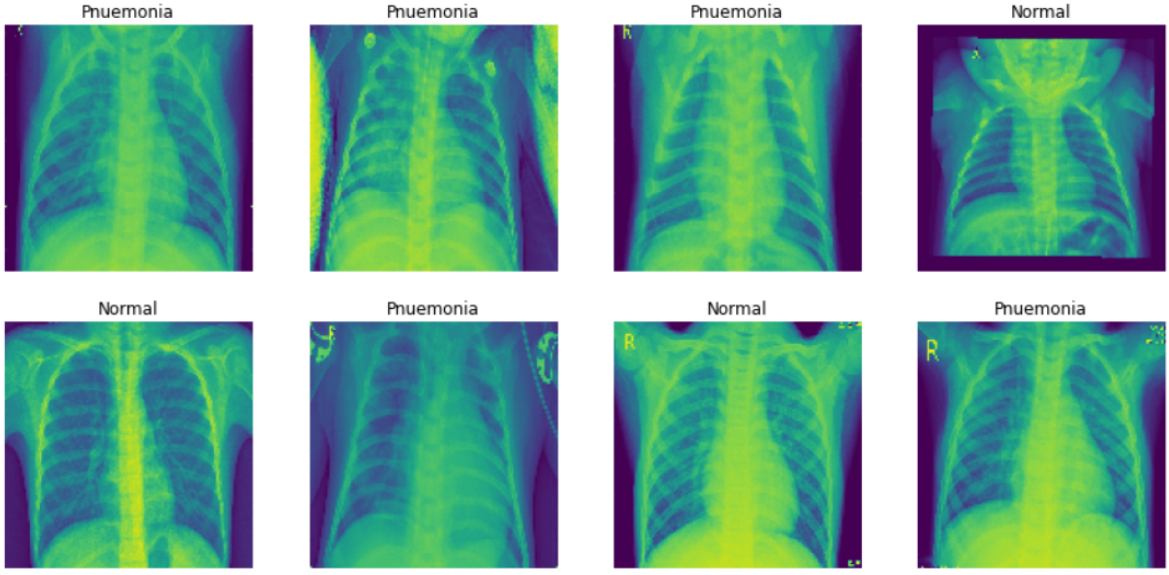


EXAMPLES

- **Training samples**



- **Testing samples**



Model Summary

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
sequential_1 (Sequential)    (None, 64, 64, 128)       1792
_________________________________________________________________
sequential_2 (Sequential)    (None, 32, 32, 32)        37024
_________________________________________________________________
sequential_3 (Sequential)    (None, 16, 16, 32)        9376
_________________________________________________________________
flatten (Flatten)            (None, 8192)              0
_________________________________________________________________
sequential_4 (Sequential)    (None, 64)                524608
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 65
=================================================================
Total params: 572,865
Trainable params: 572,353
Non-trainable params: 512
_________________________________________________________________
None
```

Model Performance

- **Initially**

  Epoch 001: Loss - 1.47 | accuracy - 91.52 | val_loss - 3.48 | val_accuracy - 50.00
  Epoch 002: Loss - 0.43 | accuracy - 85.52 | val_loss - 2.90 | val_accuracy - 50.00

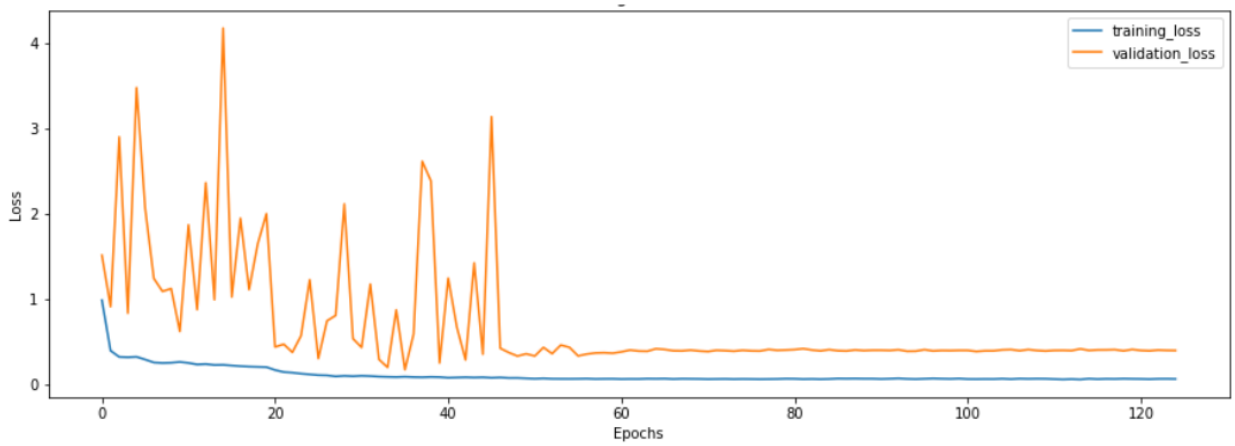
- **Finally**

  Epoch 124: Loss - 0.06 | accuracy - 98.76 | val_loss - 0.39 | val_accuracy - 93.75
  Epoch 125: Loss - 0.06 | accuracy - 99.07 | val_loss - 0.39 | val_accuracy - 93.75

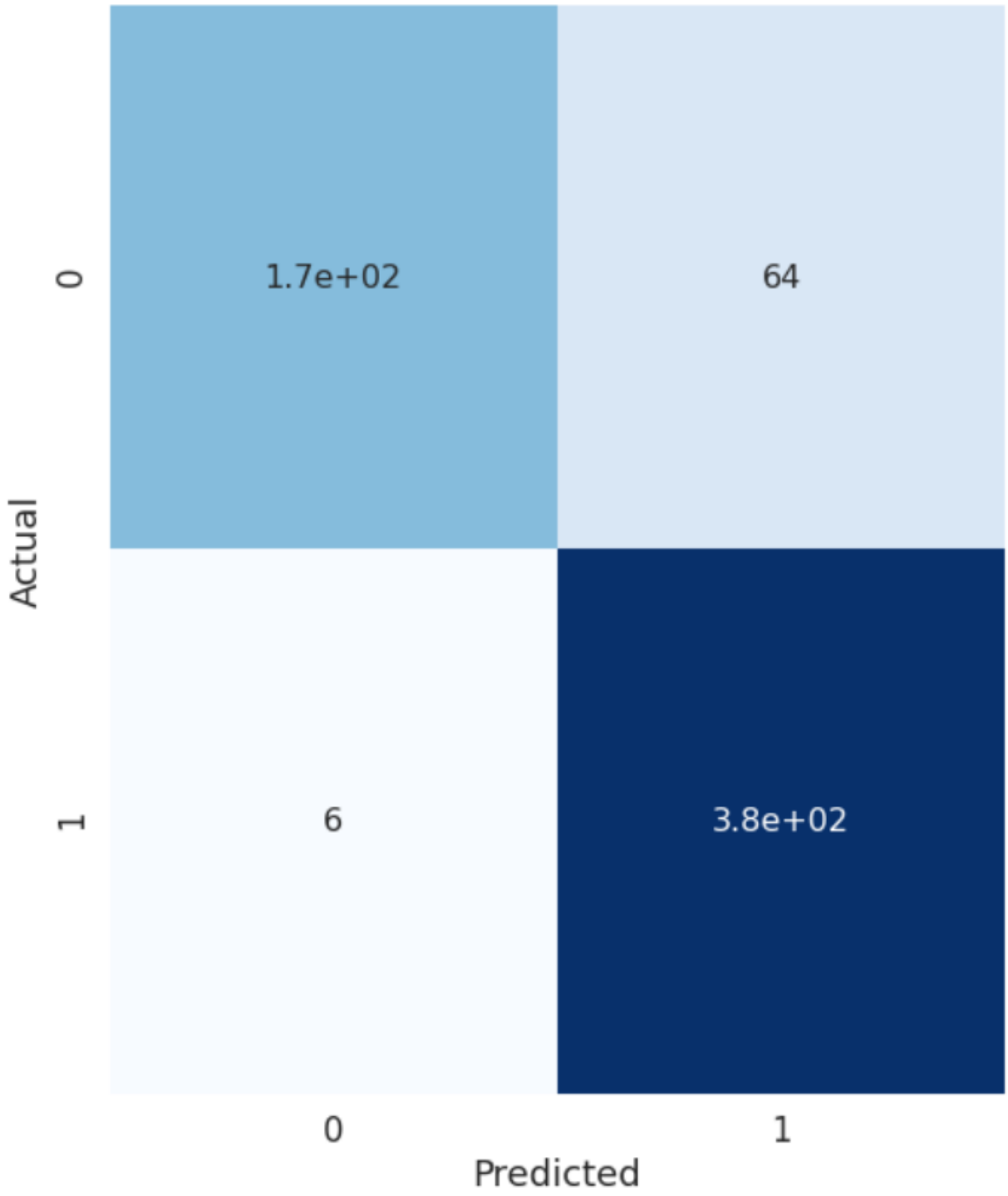
PERFORMANCE

Accuracy

Losses



Confusion Matrix



Link to the code - Kaggle notebook