Lung Cancer Detection Using CNN

Mohammed Tahir
School of Computing Science Engineering
Galgotias University
Greater Noida,India

tahiransr96@gmail.com

Abstract

The recent surge of Deep Learning has led to breakthrough advancements in almost every field of its application. A particular deep learning architecture, arguably the most popular one is the Convolution Neural Networks. The interest in convnets has seen an exponential increase due to their effectiveness and scalability. CNNs have become the go-to solution for image data problems and has provided results that are at par with if not better than human standards. The simplicity of the CNN architecture is another big factor of its success. The image processing and classification capabilities of CNN have found great usage in medical field, making it possible to detect and classify diseases as severe as Cancer effectively for the sake of better care. In this project, I've initiated an elaborate study of Convolution Neural Networks, built multiple architectures from scratch and furthered our understanding with the preparation of an elementary dog-cat CNN classifier model followed by a more extensive CNN model for detection of lung cancer in a patient. The project is built on Google's interactive and versatile cloud platform for AI development Google Colaboratory, using the open-source neural network library 'Keras' for model development and libraries such as matplotlib and tensorboard (tensorflow) for result plotting and analysis. Data for training and testing our model was extracted from the ' LUNA 2016 medical image database '. The model was tuned using Grid-Search and achieved over 97% test accuracy in its final iterations. To culminate,I have enlisted some future-work prospects like De-convolution/Translated-Convolution,implement one or more named CNN networks like Inception or Alexnet, test the model on larger images etc.

Keyword Lung cancer detection, image generation,deep learning, Keras,maxpooling,TensorFlow and many more.

INTRODUCTION

The use of Convolutional Neural Networks can be traced back to the nineties for character recognition purposes(LeCun et al., 1997) but it wasn't until the 2012 AlexNet that it grew to the widespread acclaim that we know of today. In less than a decade, researchers have progressed from single-digit layers in a CNN model to triple-digits, integrating various other data science techniques to create umpteen possible configurations. The architecture and working of a Convolution Neural Network can be understood better by a simple binary image-classification model such as a dog-cat CNN that classifies an image as one of the two preset classes, Dog or Cat. The architecture of CNN models is predominantly similar, beginning with a few convolution layers that apply various convolution filters or kernels or masks to the input image. An image is represented as a matrix of gray-scale or color intensity values at the pixel represented by the matrix cell indices. Each filter/kernel can be thought of as a feature extractor that extracts positions of that particular feature represented by the kernel, on the input image. The convolution operation hence produces a feature map .
The convolution layer is each followed by a pooling layer generally that sub-samples an image to provide lower dimensional matrices for better computations. A series of convolution + pooling layers are followed by a number of densely or fully connected layers after flattening the output. The fully connected layers operate as a typical neural network and finally classifies to binary (sigmoid/relu) or multiple classes (softmax). The medical field is a likely ground for machine learning practice and application, as medical regulations allow increased sharing of anonymized data for the sake of better care. The field is pretty young and flourishing at the forefront of technology. This problem is interesting and promising in that it has impactful implications for the future of healthcare, deep learning applications affecting personal decisions, and computer vision in general. Cancer is a disease that needs no documentation of its perils. It could be present in almost any part of the body like lungs, breast, prostate, skin etc. Lung cancer is the leading cause of cancer-related deaths worldwide. The National Lung Screening Trial (NLST), a randomized control trial in the U.S. including more than 50,000 high-risk subjects, showed that lung cancer screening using annual low-dose computed tomography (CT) reduces lung cancer mortality by 20% in comparison to annual screening with chest radiography. Due to the absence of any true remedy or cure for the disease, detection in the early stage is crucial in preventing its spread. In this project, I am interested to train a convolution neural network and then use the trained model to solve a binary image classification problem to classify as positive or negative results.

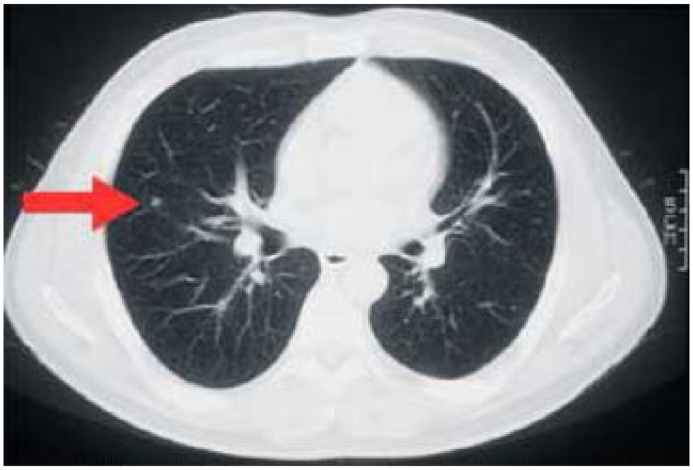


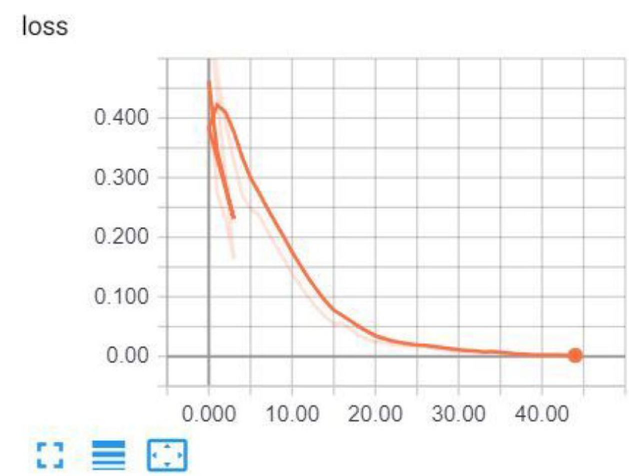
Figure 1: 2D CT scan slice containing a small (5mm) early stage lung cancer nodule

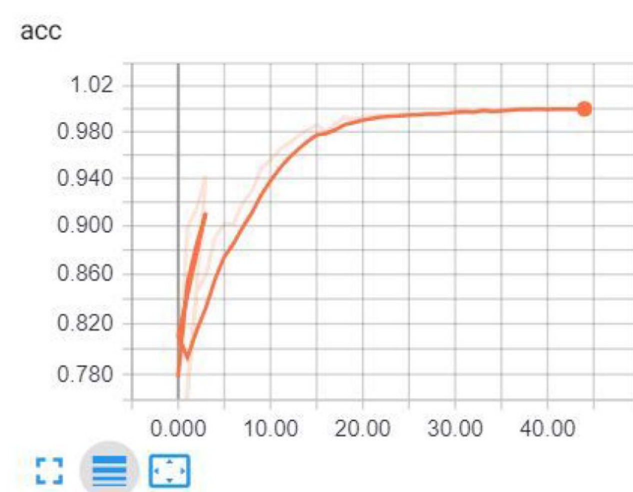
METHODOLOGY

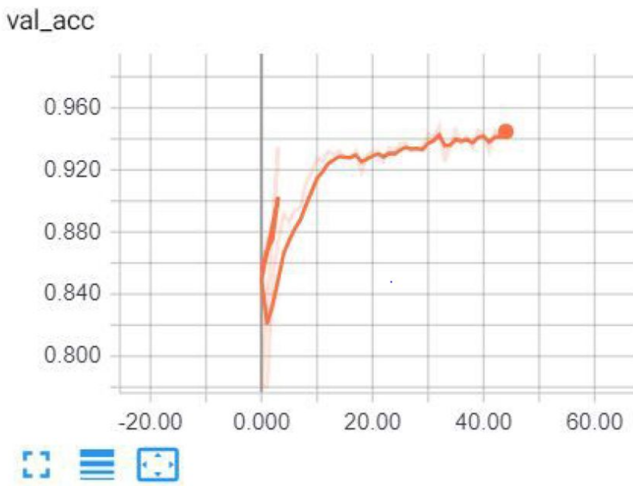
The Goal of the project is to train a Convolution Neural Network for binary image classification i.e detecting whether a person has lung cancer or not , which includes following tasks:
● Fetch and Preprocess the whole dataset.
● Train a Neural network on the training data.
● Optimize the network to improve accuracy.
● Compare the performances of Algorithms used.
Inputs: The inputs to the network are 40*40 pixel image snippets.
Outputs: The output of the network is the class label for the input image.
Data source:
Publicly available LIDC/IDRI database is used. This data uses the Creative Commons Attribution 3.0 Unported License . Scans with a slice thickness greater than 2.5 mm were excluded. In total, 888 CT scans are included.Each radiologist marked lesions they identified as non-nodule, nodule < 3 mm, and nodules >= 3 mm.
Data Description :
Total 2948 images are used.
The data includes three datasets:
● Training Data: Contains total 2064 grayscale images
● Validation data: Contains 442 grayscale images
● Testing data: Contains 442 grayscale images
Image size 40*40 pixels
Model:
The model is implemented using the Sequential API in Keras. It consists of three Convolution-2D layers with 16, 32 and 64 kernels of size (3,3), (5,5) and (7,7) respectively each followed by a Maxpooling-2D layer of size (2,2) pooling with a stride of 2. Maxpooling layers reduce the dimensionality of data thereby reducing the number of parameters, leading to a reduction in training time and tackling overfitting. Pooling layers downsample each feature map independently. They reduce the height and width, keeping the depth constant. In the end i have fully connected layers that flatten the last convolution layer's output and connect every node of the present layer with the other node of the next layer. Neurons in a fully connected layer have full connections to all activations in the previous layer, as they have in regular Neural Networks. I need the Flattening layers since the output of both convolution and pooling layers are 3D volumes, but a fully connected layer expects a 1D vector of numbers. The Flattening operation simply involves arranging the 3D volume of numbers into a 1D vector. The flatten output becomes an input to the fully connected layers. I then implement grid search to find the best values for hyperparameters optimizer, epochs and batch size.

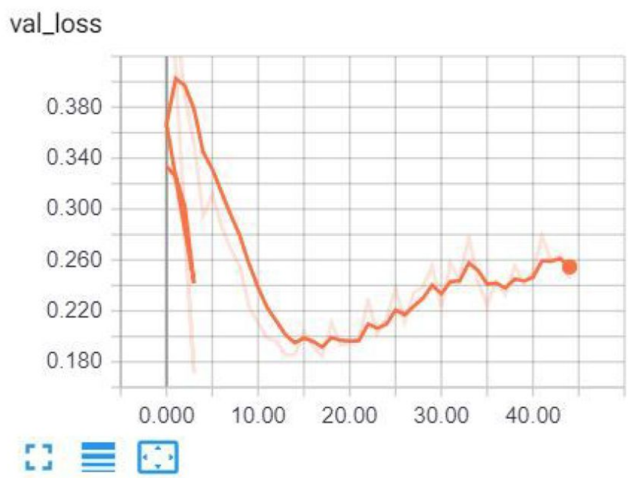
Results

The best combination of parameters calculated from grid search i.e parameters of 7th an experiment was used to train the model. The model achieved an accuracy of 97.51%. The final loss and accuracy graphs for training and validation have been plotted using tensorboard as below:









Acknowledgement

I would like to acknowledge my guide Dr. Kuldeep Singh Kaswan for providing me the necessary guidance and valuable support throughout this research project which also helped me in doing a lot of Research and I came to know about so many new things. I am really thankful to him. I would also like to thank my family and friends for their guidance and assistance throughout the project.

CONCLUSION

In this project, I've explored the very powerful Convolution Networks and applied it to predict the presence of lung cancer in a patient using an image snippet of a scan. Experimentation showed that CNN was highly efficient in performing the task and the accuracy was further improved by optimization of hyperparameters. The images used specify a predetermined The section of the lung and the model is also rather simple but still I was able to get ground-breaking results.
I plan to explore Deconvolution, Transpose-Convolution for Upscaling and other possible extensions of CNN. For the Cancer model, i would like to implement one or more of the named CNN networks like Inception or Alexnet using transfer learning and comparing the workings and result with the previous models. The current model works on images of 40 x 40 dimensions, increasing which might be plausible to perform testing on larger scans, possibly of the entire lung instead of a predetermined section.

References

- https://apollack11.github.io/machine learning.html
- https://keras.io/
- https://www.tensorflow.org/
- https://www.healthline.com/health/lung-cancer#stages
- https://towardsdatascience.com/building-a-convolutional-neural-network-cnn-in-keras-329fbbadc5f5
- https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/