Convolutional Neural Networks in Radiology

Max Norris mpnorris@stanford.edu | CS 230 Fall 2018 Course Project

Introduction

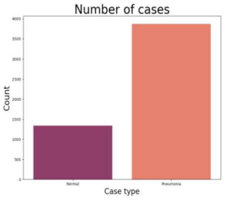
Motivation: The goal of using CNNs in Radiology is to help Radiologists more accurately diagnose patients, to reduce fatigue-based errors, and to help patients who lack access to Radiologists diagnose themselves using their chest X-Ray images. My CNN specifically checks for pneumonia, which can cause coughs, fevers, shortness of breath, chills, shaking, fatigue, sweating, and muscle pain, so it would be excellent to have a fast, accurate method for detecting such a disease before symptoms worsen.

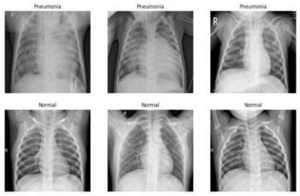
Approach: I used partial transfer learning from VGG16 and a 24-layer CNN and I split my image data into a training set (N normal = 1341, N pneumonia = 3875), a cross validation set (N normal = 8, N pneumonia = 8), and a test set (N normal = 234 normal, N pneumonia = 390).

Results: Pneumonia is not easy to see with the naked eye—when interpreting an X-Ray, a professional Radiologist will look for white spots in the lungs (called infiltrates) that identify an infection. Nevertheless my best model achieved a F1-score on the test data of 0.86.

Data and Features

- 5863 infant X-Ray images from Guangzhou Women and Children's Medical Center in China
- Every image was labeled with the ground truth (pneumonia vs. normal)
- I performed data augmentation (horizontal flips, rotation, random brightness changes) to correct for class imbalance (as seen below there are more pneumonia examples)
- The image data varied in dimensions, so I converted all images to a standard 224x224x3—the standard input for the VGG16 CNN.





Models

- Rather than using random initialization for my first four convolution layers, which capture general details like blobs, patches, edges, etcetera, I found much more success through loading pre-trained weights from VGG16 (built by Visual Geometry Group) and fine tuning them.
- Next I use convolutional, batch norm, and max pooling, layers, ending with a softmax layer for binary prediction. I use a 24-layer CNN instead of the 16-layer VGG16 CNN and update my parameters according to Adaptive Moment Estimation (Adam)
- I use binary cross entropy as my loss function $BCE = -\frac{1}{N}\sum_{i=0}^{N} y_i \cdot log(\hat{y}_i) + (1-y_i) \cdot log(1-\hat{y}_i)$
- I experimented with different weight initializations, different learning rates, and different batch sizes.

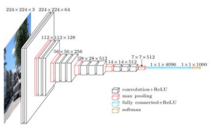
$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$
$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

Adam

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t$$

Note : default values of 0.9 for β_1, 0.999 for β_2, and 10^{-8} for ϵ



Results

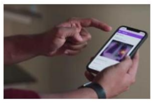
Training: N Pneumonia = 3875, N Normal = 1341

Testing: N Pneumonia = 390, N Normal = 234

Weight Initializations	Batch Size	Learning Rate	Effective Error Train	F1 Score Train	Effective Error Test	F1 Score Test
Random	16	1x10^-3	0.50	0.85	0.50	0.77
Random	16	1x10^-4	0.50	0.85	0.50	0.77
VGG16	16	1x10^-3	0.50	0.85	0.50	0.77
VGG16	16	1x10^-4	0.01	0.99	0.16	0.86
VGG16	64	1x10^-3	0.50	0.85	0.50	0.77
VGG16	64	1x10^-4	0.01	0.99	0.16	0.86

Discussion and Future

The CheXNeXt team at Stanford, which includes Professor Andrew Ng and CS 230 guest speaker Pranav Rajpurkar, has developed a CNN that classifies 14 different thoracic diseases roughly as well as the top Radiologists. Additionally, using class activation mappings (CAMs), their model outputs a heatmap pinpointing the location in the chest that caused the CNN to make its prediction, as seen in their colorized example below. Now they are working to deploy a website where anyone in the world who lacks access to a Radiologist can upload their own X-Ray image file to get a fast, free diagnosis. With six more months, I would work to try to replicate or improve upon their model. Since their results were so promising, I expected my top model to achieve a high F1 Score as well.



	F1 Score (95% CI)
Radiologist 1	0.383 (0.309, 0.453)
Radiologist 2	0.356 (0.282, 0.428)
Radiologist 3	0.365 (0.291, 0.435)
Radiologist 4	0.442 (0.390, 0.492)
Radiologist Avg.	0.387 (0.330, 0.442)
CheXNet	0.435 (0.387, 0.481)

Acknoledgements and References

I would like to thank my project mentor, Cristian Bartolomé Aramburu, for his guidance throughout my project during my proposal and milestone. I would also like to thank Sarah Najmark for her excellent instruction during Friday TA Sections and the other CS 230 TAs, especially Steven Chen, for their time and assistance during Office Hours.

Works Cited:

[1] https://nihcc.app.box.com/v/ChestXray-NIHCC [Accessed: October 10, 2018]

[2] https://stanfordmlgroup.github.io/projects/chexnet/ [Accessed: October 15, 2018]

[3] https://aws.amazon.com/blogs/machine-learning/get-started-with-deep-learning-using-the-aws-deep-learning-ami/ [Accessed: December 2, 2018]

[4] https://www.kaggle.com/aakashnain/beating-everything-with-depthwise-convolution?scriptVersionId=4028995 [Accessed: December 2, 2018]

[5] https://www.radiologyinfo.org/en/info.cfm?pg=pneumonia [Accessed: December 10, 2018]