Applications of Transfer Learning in Pneumonia Diagnosis

CSCI-C 490 Application of Deep Learning Final Project

Joseph McCombs
Indiana University South Bend
jamccomb@iu.edu

Jakob Schuppan
Indiana University South Bend
schuppan@iu.edu

Chris Cowell
Indiana University South Bend
ccowell@iu.edu

Abstract—Transfer Learning is a deep learning technique that allows users to build accurate models very quickly. Our project set out to use transfer learning with pretrained weights and model architectures to classify a pneumonia dataset. We used a variety of model architectures with ImageNet weights. We also used weights from a pretrained Gleason score classifier to attempt to classify the dataset. We achieved varying results with all of the test cases, but it was clear that Transfer learning sped up the learning process and achieved overall better results.

Keywords—transfer learning, convolutional neural networks, deep learning, fine-tuning, gleason, Kaggle, pneuomonia

I. Introduction to Transfer Learning

Transfer learning is a highly effective approach to building a deep learning model with a pretrained network. A pretrained network was originally built and trained on a large dataset. If the original dataset was large enough and general enough, then the spatial hierarchy of the features learned from thee pretrained network can act as a generic model to be used in different computer vision problems (Chollet). ImageNet was trained on millions of images with more than 20,000 classes and the weights are readily available to use in Keras. The images may not be directly relatable to the task that the user is attempting to accomplish, but the dataset was large enough and general enough that the features could be applied to other applications. These image weights will be loaded into a convolutional base, which is used to extract features for the classifier. Keras comes preloaded with models that can be used for the convolutional base which include Xception, Inception V3, ResNet50, VGG16, and MobileNet among others that can be used. Since some of these model architectures contain upwards of 16 million parameters, a certain amount of the layers must be frozen in order to preserve the weights that are loaded into the convolutional base; training 16 million would also be computationally expensive. A classifier is built on top of the convolutional base, much like typical convolutional neural networks, which are always trainable parameters. The layers in the convolutional base which are not frozen are considered layers that you are fine tuning. It is typically a good general strategy to fine-tune the top 2-3 layers of the convolutional base (Chollet). In our use case, the strategy used is to remove the fully connected layers near the bottom from being trainable and introduce a new classifier. This allows the entire imported set of weights to mutate as needed to fit the new dataset. Since the dataset we are applying to the model is not very different from the one it was originally trained upon, drastic differences are not expected, and the slight mutations are desired to fit the nuances in the new image data.

II. Gleason Score Model

Now with transfer learning, we can apply any previous learning from other models into another of our choice. The base model we will use for these are from a convolutional network trained to identify the Gleason score of prostate cancers. This model itself was trained using a convolutional base of MobileNet and weights from ImageNet. Essentially the weights are highly modified, fine-tuned version of ImageNet repurposed for classifying the gleason score of prostates. This is acutely ideal for our application because this is trained to look at two dimensional DICOM images converted to JPEG , precisely the same as our project.

III. Pneumonia

A. About the Data

The data used is a collection of converted DICOM to JPEG images separated into two categories: pneumonia and normal. Daniel Kermany, et al, collected this data from various pediatrician x-ray labs; 5,232 in total. These images were then labeled by three tiers of graders - the first were an initial layer of quality control that removed images containing severe artifacts or images of poor resolution. The second tier consisted of four ophthalmologists working independently to grade these images. Finally, a third tier of two senior retinal specialists verified the true labels for these images. It is important to note that of the data labeled 'pneumonia', they are contracted in two different ways being either bacterial or viral. The other 1,349 chest x-rays are of children with no lung, or chest condition whatsoever.

The data is split into three categories: test, train and val. The train set contains 1349 images of normal chest x rays and 3883 images of chest x-rays with pneumonia, which is used to train our models. The test set contains 234 images of normal chest x-rays and 390 images of chest x-rays with pneumonia, which is used for validation purposes. We would expect guessing to be around 60% accurate due to this. The val set contains 8 images of normal chest x rays and 8 images of chest x-rays with pneumonia, which were unused in our project.

B. Kaggle Reproduction

We applied our transfer learning model by replicating what had been done by the best scorer on Kaggle so far. Rohit Verma's repository implemented InceptionV3's convolutional base model and weights to win the top score on Kaggle. Our first goal was to replicate this as close as possible to give us the best chance at success. After using his self-written tutorial on Github, we were able to get a few successful runs on the pneumonia data set.

The first replication we performed was of his pneumonia diagnosis model that did not transfer learning. The intention behind this was to demonstrate the substantial difference in validation accuracy and loss that this method allows. The replicated model that did not use transfer learning for pneumonia diagnosis was able to attain a validation accuracy and validation loss of 70.67% and 0.7377 respectively after 6 epochs. Beyond that, overfitting was observed and could not reach anything better.

After observing the first model without transfer learning scores, we saw room for improvement in the transfer learning replication. This model involved an immense amount of parameters comparatively speaking; 532,738 to 22,065,826. After only two epochs, we saw a validation accuracy and validation loss of 86.38% and 0.3406 respectively; much improved over the isolated version.

C. Gleason Weights With MobileNet's Convolutional Base

After having a model successfully train on our data using the Kaggle winner's setup using InceptionV3's base model and with his classifier, we began to swap out the convolutional base model for MobileNet's and the InceptionV3 weights for the weights generated by the Gleason Scoring project. Essentially we kept the classifier of the Kaggle winner, replaced the convolutional base with MobileNet, and imported weights with those found from the Gleason Scoring project.

This produced better results than no transfer learning at all, with 86.38% validation accuracy versus 70.67% validation accuracy after 5 epochs where it began to overfit. Though this did not trump the scores attained via InceptionV3, this does show credence to using models trained on similar data with similar goals.

D. VGG16

Although the goal of our project was already achieved by correctly implementing the gleason score weights into a MobileNet convolutional base and achieving decent results, we elected to compare this model to other models trained with and without transfer learning. We decided to train a model using VGG16's convolutional base from scratch and using the ImageNet weights for comparison.

The VGG16 model that we trained from scratch had 15.7 million parameters to train over the course of 50 epochs. The model began to overfit around 30 epochs, but achieved 91% validation accuracy and .216 validation loss. The VGG16 model trained with ImageNet weights had 8.2 million parameters to train over the course of 35 epochs. The model began to overfit around the 15th epoch, but achieved 93.75% validation accuracy and .150 validation loss.

Both models achieved impressive results, but the time consumed to train to achieve these results were drastically different. The model with ImageNet weights achieved a validation accuracy of 92.7% by the second epoch in about 5 minutes of training. The model trained from scratch performed worse in general but took 35 epochs to achieve the best results. This model took considerably longer to train considering the number of parameters that it was training and also did not perform as well as the transfer learning model.

E. Xception and ResNet50

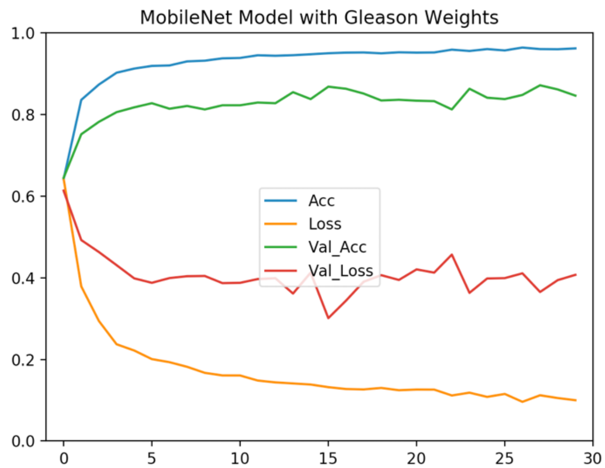
After training a model using the VGG16 convolutional base with great success, we wanted to experiment and use ResNet50 and Xception model architectures to compare the results. We still elected to use the ImageNet weights in the model, but both of these models performed poorly and began to overfit almost immediately. We experimented with different classifiers on top of the model, but did not achieve good results in any of the runs.

IV. Transfer Learning Applications

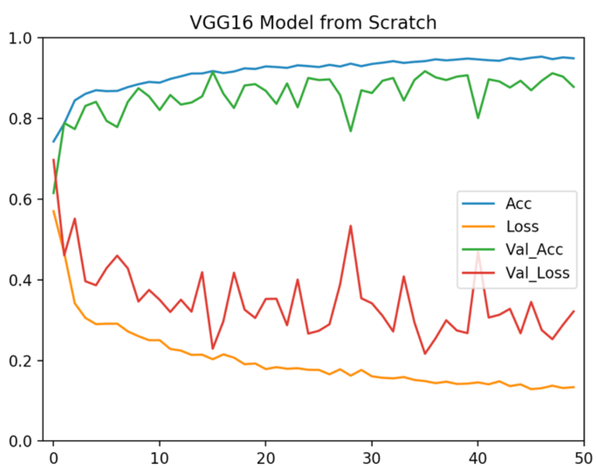
In the real world, transfer learning is commonly applied as a technique for machine learning when data is scarce or expensive to obtain. In practice, this typically means using weight data from an already trained convolutional network and then applying a feature extractor to this pre-trained network designed for the planned task. In our case, this pre-trained network was a highly modified ImageNet that is specialized in two dimensional x-rays. In our use-case, all we had to do was remove the fully connected layers near the end and use a classifier that fit our output needs.

V. Figures

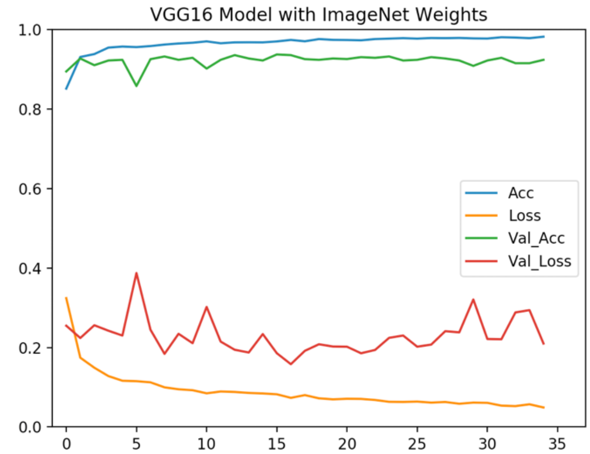
A. *MobileNet Convolutional Base with Gleason Weights*



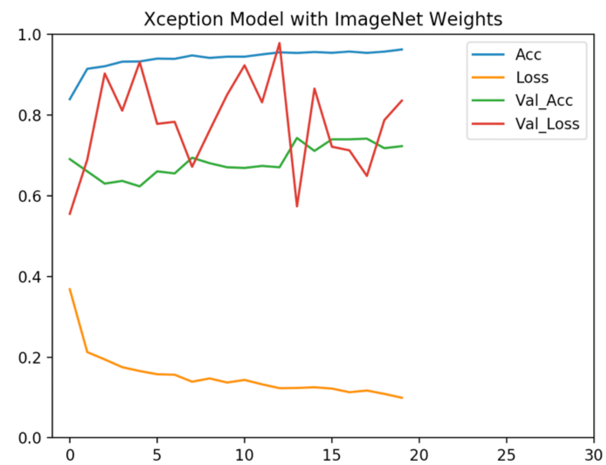
B. *VGG16 Convolutional Base from Scratch*



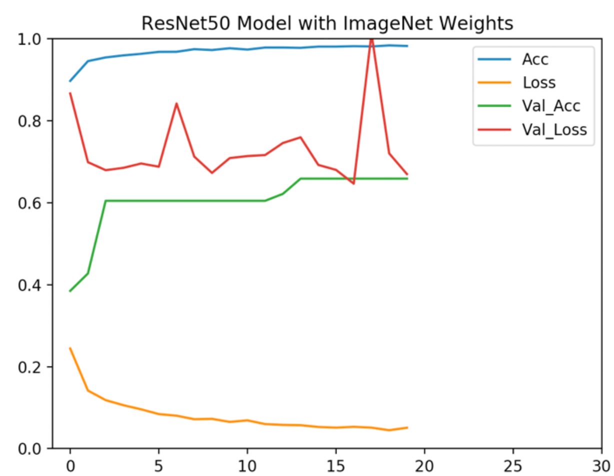
C. *VGG16 Convolutional Base with ImageNet Weights*



D. *Xception Convolutional Base with ImageNet Weights*



E. *ResNet50 Convolutional Base with ImageNet Weights*



VI. References

Arvaniti, Eirini. “gleason_CNN.” Jul, 2019. https://github.com/eiriniar/gleason_CNN

Chollet, François. *Deep Learning with Python*. 1st ed., Manning Publications Co., 2018.

Kermany, Daniel S, et al. “Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning.”, Cell, 22 Feb. 2018, www.cell.com/cell/fulltext/S0092-8674(18)30154-5. DOI: https://doi.org/10.1016/j.cell.2018.02.010

Verma, Rohit. “Pneumonia-Diagnosis-using-XRays-96-percent-Recall.” Feb, 2018. https://github.com/deadskull7/Pneumonia-Diagnosis-using-XRays-96-percent-Recall