Artificial Neural Networks (ANNs) Applications

Andrew Perez-Ledo
University of Florida
Gainesville, United States
andrewperezledo@gmail.com

Abstract- This paper details the implementation of artificial neural networks (ANNs) using TensorFlow for classification and segmentation tasks on custom flower species and lung x-ray datasets. The project aims to achieve optimal model performance through hyperparameter tuning, transfer learning, and best training practices.

I THE DATA

In this project, we have two datasets for two different classes to implement ANNs. There is a custom flower dataset of 1678 300x300 pixel images of flowers found by students at the University of Florida (UF). This dataset is for the classification of ten different flowers such as roses, magnolias, firebush, etc. Refer to figure 1.

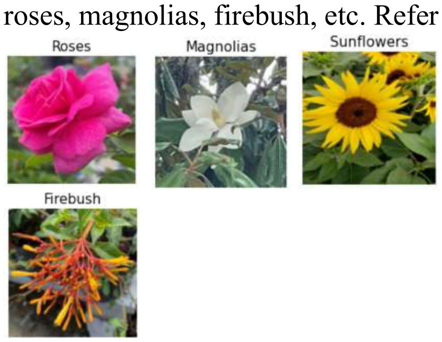

Figure 1, Example flower images.

The Lung Segmentation Dataset consists of 704 chest X-ray images and their corresponding lung segmentation masks. Each image is grayscale and has a dimension of 512x512 pixels. The dataset is further divided into a training set of 566 images and masks, and a test set of 138 images and masks. The objective is to train a neural network to accurately segment the lung regions in the X-ray images based on the provided ground truth masks (segmentation labels). Refer to figure 2.

Figure 2, Example x-ray & mask images

II THE GOAL

The overall goal of this project is to utilize TensorFlow for image analysis tasks. We'll build and train artificial neural networks (ANNs) for flower species classification and lung X-ray segmentation. Hyperparameter tuning, transfer learning, and best training practices (checkpointing, early stopping, adaptive learning rate) will be implemented to optimize model performance. We'll evaluate the trained models for real-world applications in agriculture and medical imaging, while also addressing limitations and proposing mitigating strategies.

III ANNS FOR FLOWER CLASSIFICATION

We will experiment with multiple different types of ANN architectures including multi-layer perceptions (MLPs) and convolutional neural networks (CNNs).

It is important to note that accuracy of the classification will be used for the metrics we gauge our models' effectiveness.

The first network being experimented with was a simple sequential MLP that was shown in lecture featuring two hidden layers with 300 and 75 neurons each. Both used the rectified linear unit (RELU) activation function and a SoftMax output activation function with 10 units due to the need for probabilities of the images being one of the 10 classes of flowers. Additionally, it also used a sparse categorical entropy loss function due to the categorical nature of our flower's classes and the NADAM optimizer for its use of adaptive learning rate and Nesterov accelerated gradient. It is important to note that all models went through experimentation with more hyperparameters such as batch size and number of epochs. This model resulted in an overfitted performance when comparing training and validation sets. Refer to figure 3.

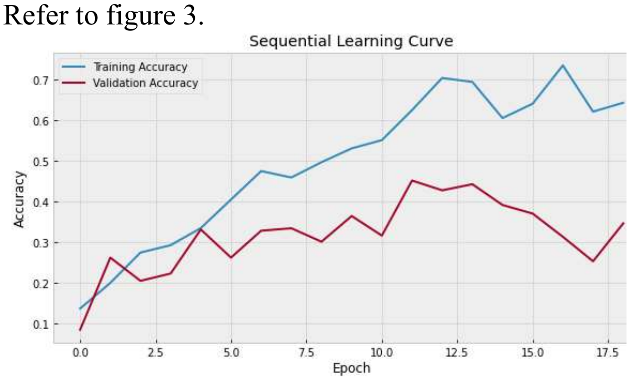

Figure 3, Learning curve of simple network

The next few models all consist of classical architectures for transfer learning as recommended for the limited dataset.

In our exploration of transfer learning architectures for the flower classification task, we experimented with Xception, ResNet 152 V2, and MobileNet. While Xception achieved the lowest performance and exhibited the most significant overfitting, ResNet 152 V2 offered some improvement in overfitting but struggled more with overall accuracy. MobileNet emerged as the most effective model, demonstrating the best performance among the three. However, it's important to note that even MobileNet exhibited some degree of overfitting, suggesting further hyperparameter tuning or data augmentation techniques might be necessary for optimal results. Refer to figure 4, 5, and 6.

With the initial attempt of Xception, many routes were taken to attempt to reduce the overfitting such as increasing droupout rate, including pooling and upsampling layers, and L2 regularization.

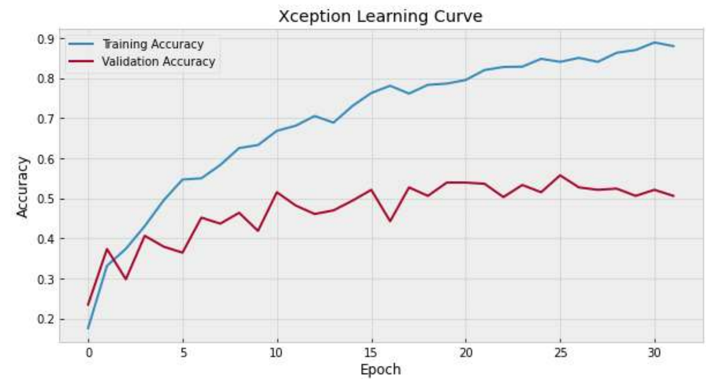

Figure 4, Learning curve of Xception

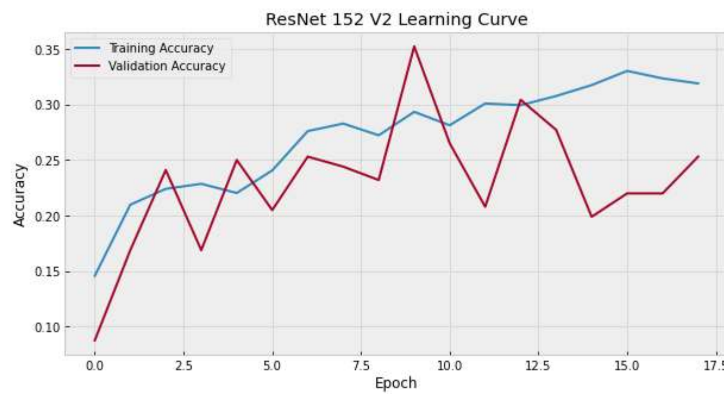

Figure 5, Learning curve of ResNet 152 V2

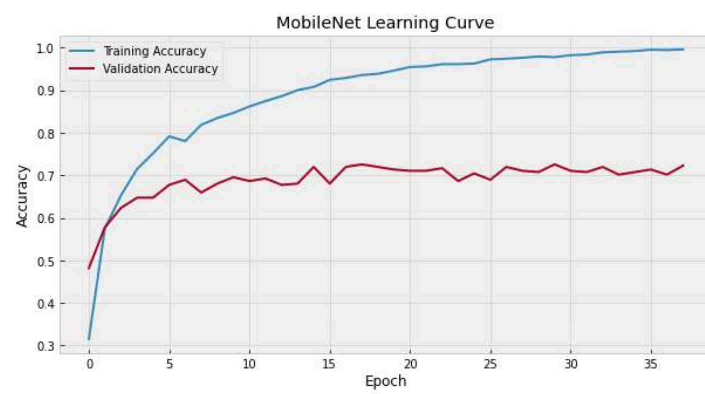

Figure 6, Learning curve of MobileNet

One thing to take into consideration about these architectures is their difference in number of parameters which could have had a large impact on their performance for this data. Xception has around 22 million parameters, ResNet 152 V2 is the largest with around 60 million parameters, and MobileNet was the smallest with only just more than four million parameters. Based on each model's performance, the two larger architectures were most likely too complex for the task at hand making the simplest network MobileNet prevail.

IV ANNS FOR LUNG SEGMENTATION

We investigated two deep learning architectures for lung segmentation: Multilayer Perceptron (MLP) and Convolutional Neural Network (CNN). Both approaches also used NADAM optimizers as well as a binary cross entropy loss function because the aim of this is classify each pixel in the image as the lung or not the lung resulting a series of binary classifications that combine into a segmentation. Furthermore, both also had only a single neuron for the top layer which used the sigmoid activation function which provided the probability of each pixel of being part of the lung.

As a baseline, we started with a simple MLP with several densely connected layers. While achieving decent results, it struggled to capture the spatial relationships between pixels crucial for accurate segmentation. Additionally, smaller MLP architectures performed significantly worse, highlighting the limitations of MLPs for image analysis tasks. However, one of the most prominent problems with the MLP was the fact that the network would misclassify non-lung pixels and lung pixels such as the outlines of the shoulders and torso.

Due to its ability to learn spatial features, a CNN emerged as the superior choice. The CNN architecture included average pooling, convolutional layers with varying filters and activation functions, and upsampling to recover the original image size. Hyperparameter tuning played a crucial role in optimizing the CNN's performance. Tuning parameters like the number of layers, filters, strides, and batch size helped us achieve accurate lung segmentation for both the training and validation set while avoiding overfitting to the training data. Notably, the CNN successfully avoided misclassifying areas outside the lungs, resulting in more precise segmentation masks. Refer to figures 7-10.

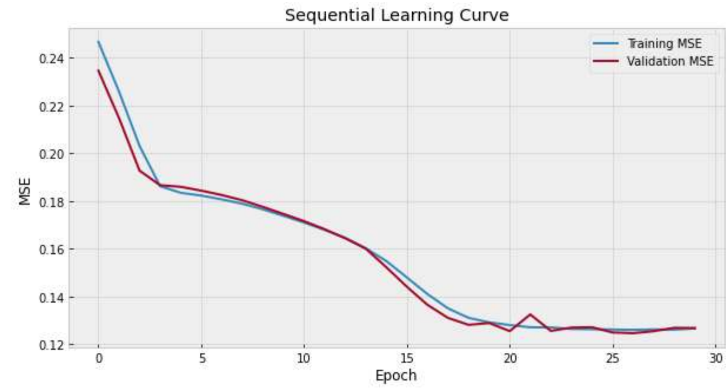

Figure 7, Simple MLP Learning curve

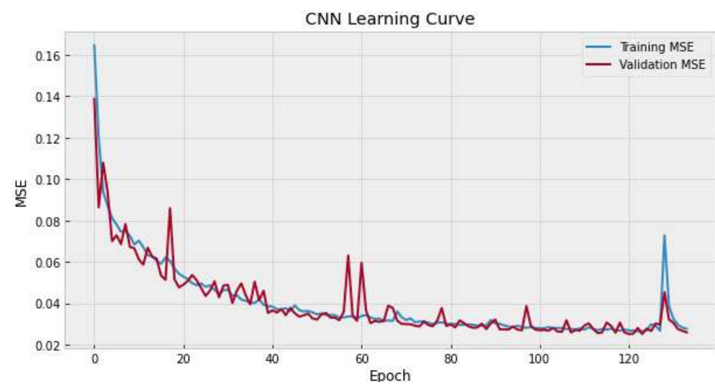

Figure 8, CNN Learning curve

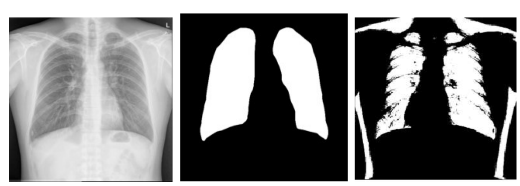

Figure 9, MLP image, truth mask, and prediction mask

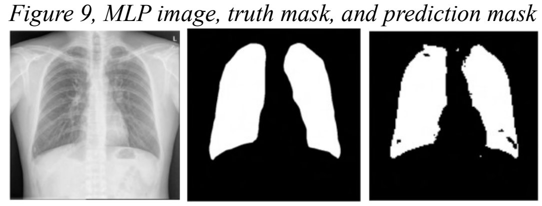

Figure 10, CNN image, truth mask, and prediction mask

<u>V TEST SET</u>

Despite promising performance on the validation sets, both the flower classification and lung segmentation models exhibited a drop in accuracy when evaluated on the unseen test data. The flower classification model's accuracy decreased to 68%, potentially due to the presence of misleading images with unbloomed flowers in the test set as well as some *incorrectly labeled images*. Similarly, the lung segmentation model struggled with the test set, resulting in incomplete lung segmentations with random patches misclassified near the actual lung regions. While the model seemed to capture the general boundaries of the lungs, it left significant unfilled spaces within the lung area and produced pixelated regions around the lungs that were mistakenly classified as lung tissue. This highlights the limitations of the models and the importance of incorporating data augmentation techniques or exploring more robust architectures to improve generalizability on unseen data. The increase in MSE (0.159) for the CNN network and the rise in loss functions for both models further prove the challenges faced by the models when encountering data outside the training distribution. Refer to figures 11-12.

Figure 11, Examples of Misclassified images on (According to labeled test data) ("Lillies" image is mislabelled in data)

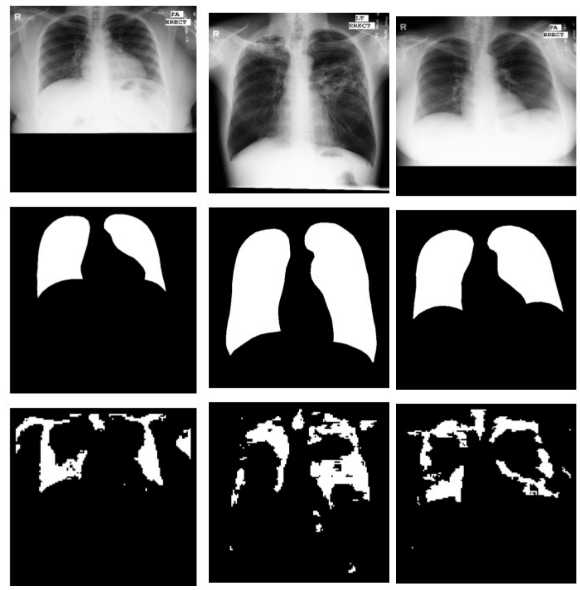

Figure 12, Examples of CNN Segmentation image reconstructions on test data