Chest X-Ray Pneumonia Detection Utilizing Machine Learning

Authors: Jeremy Lau, Shasta Narayanan, Sungjun Park

DS340: Introduction to Machine Learning and Artificial Intelligence

Professor Kevin Gold

Introduction

Pneumonia is a lung infection that can be caused by bacteria, viruses, or fungi. It is a common disease, especially in children and the elderly. The symptoms of pneumonia tend to include fever, cough, shortness of breath, and chest pain. In severe cases, pneumonia can lead to hospitalization or even death. Chest X-rays are a common way to diagnose pneumonia, however, it can be difficult for doctors to distinguish between pneumonia and other lung diseases simply using this technology. Accurate and timely diagnosis of pneumonia is therefore critical for effective treatment and management of the disease. This is where machine learning can help. Machine learning models can help automate and standardize the interpretation process, reducing the potential for human error and improving diagnostic accuracy. In this paper, we will use machine learning to create and evaluate models that can recognize pneumonia using chest X-rays.

The dataset used in our experiment contains chest X-ray images of pediatric patients from Guangzhou Women and Children's Medical Center in Guangzhou. The selected images are of patients aged one to five years old, as pneumonia is particularly prevalent in this age group. The dataset includes both images of patients with and without pneumonia, allowing us to train our machine learning models to differentiate between the two.

In our experiment, we used two different machine learning models to classify the X-ray images as either showing signs of pneumonia or not. The first model we used was a k-nearest neighbor mode (KNN). This is a non-parametric machine learning algorithm that is commonly used in classification tasks. KNN models are a type of machine learning algorithm that can be used to find the most similar examples in a dataset. The second model we used was a convolutional neural network (CNN) using Keras, which is a deep learning framework widely used in image classification tasks.

We train our models on a dataset of chest X-rays that have been labeled with pneumonia or normal. We then evaluated our models on a test set of chest X-rays that have not been labeled. We believe that our machine learning models can be used to improve the diagnosis of pneumonia. As a result, this could lead to earlier diagnosis and treatment, which could improve patient outcomes.

Methodology

CNN Hyperparameters tested:

- Combinations of random flipping, rescaling, rotation, and gaussian noise
- Number of convolutional and max pooling layers
- Number of dropout layers
- Dropout layer values
- Dropout layer placement
- Total number of training epochs
- Consecutive convolutional layers before a max pooling layer
- Different sizes for our first layer
- Different optimizers (adam, rmsprop)

We began by using the most simple neural networks that only used convolutional and max pooling layers. We discovered that five convolutional and max pooling layers were optimal and produced the highest validation accuracy on our datasets. We then tried to prevent overfitting by adding random flipping, rescaling, rotation, and gaussian noise to process the images before feeding them into the neural network. We discovered that random flipping, rescaling, and rotation all worked somewhat poorly when used by themselves. When we used all three of them together, we got better validation accuracy on our models. When we added gaussian noise, the models performed much worse no matter the combination of preprocessing techniques we used, so our best model does not use a gaussian noise layer to prevent overfitting.

Next, we began by adding dropout layers. We first started by adding dropout layers one by one until we had a dropout layer for each convolutional layer. We also tried adding only one dropout layer at the start or the end of the network, but this produces worse results than many different dropout layers throughout the network. From here, we played with the dropout layer values and fine tuned the model. We found that a dropout layer value of 0.05 was optimal.

We also played with the number of training epochs. We saw that sometimes we got very good results in the first few epochs, but this would likely not translate to working on a larger dataset because the model did not have enough time to learn the patterns in our data. We tried epoch values including 10, 15, 18, 20, 25. We noticed that the best number of epochs seemed to be around 18 epochs. Too few epochs did not give us enough time for the model to learn the data properly, and too many epochs made little to no changes to the performance of the model.

We also tried different optimizers and found that the adam optimizer worked best for our models. Consecutive convolutional layers before a max pooling layer worked somewhat well, but not as well as our previous models. We also tried different sizes for our initial layer of our network, where we found that the best performance was when our initial layer size was 32.

Our best performing model had a 0.9691 accuracy and a 0.8790 validation accuracy. This model had an initial layer size of 32; rescaling, random flip, and random rotation layers; a universal dropout value of 0.05; dropout layers after each convolutional and max pooling layers; five convolutional layers; 18 epochs; and utilized the adam optimizer. While we did produce models where we had a near perfect accuracy, those models heavily overfit the data and had much lower validation accuracies.

KNN Hyperparameters tested:

- PCA to reduce the amount of features, to n number of components that represent X% variance of data
- Number of neighbors

For our KNN model, we found that fewer neighbors produced better results. We also found that a PCA value that was not too large or not too small was better for modeling our data to prevent overfitting.

Our best performing model had a 0.8109 accuracy. This model had 10 neighbors and a PCA value of 0.9.

Results

	Best KNN	Best CNN
Train Accuracy	———————————	96.91%
Validation Accuracy	81.09%	87.90%

Between KNN and CNN, we found that CNN had a better overall validation accuracy. The best KNN model used few neighbors and a medium PCA variability percentage. The best CNN model utilized the adam optimizer with 18 epochs, five convolutional layers, random flip and rotation layers, dropout layers, and a universal dropout value of 0.05 with an initial layer size of 32.

<u>KNN model</u>

Parameter: Number of Neighbors

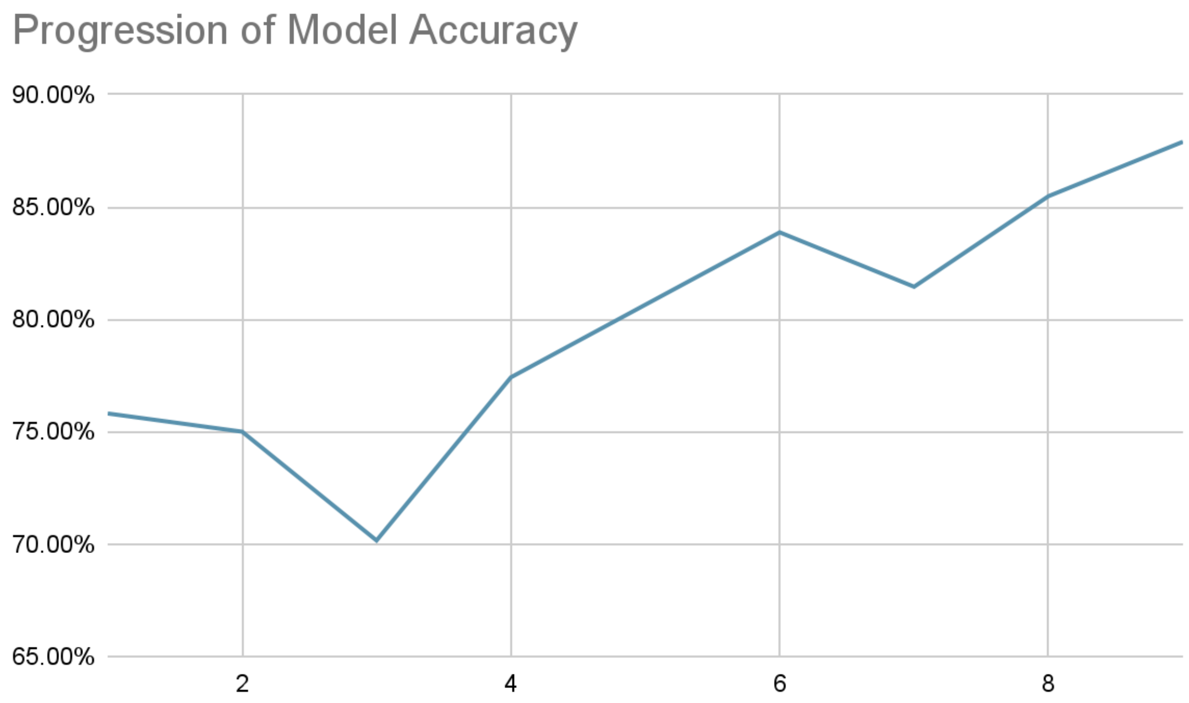
	10 neighbors	12 neighbors	15 neighbors	30 neighbors
Validation Accuracy	73.4%	73.56%	72.76%	71.15%

Parameter: PCA variability percentage*

	80%	90%	95%
Validation Accuracy	75.96%	81.09%	77.56%

*Used 10 neighbors after testing various neighbor values with PCA of 95%

CNN model



Changes (in order)

1. Adam Optimizer
2. Rmsprop Optimizer
3. Rmsprop, binary_crossentropy, extra conv2D layers and max pooling
4. Adam Optimizer, Added dropout and random rotation
5. Added rescaling, added more dropout, removed extra 512 layer
6. Increased dropout rate, slightly increased epoch
7. Increased 2 out of 3 dropouts to 0.5
8. Double CNNs for each filter, and lower filter to start form 8
9. Reduce epochs, Reduce filter size

Conclusions

Our project has demonstrated that both KNN and CNN models can accurately classify chest X-ray images to identify signs of pneumonia. This experiment has taught us about the importance of optimizing hyperparameters and adjusting the tools to avoid overfitting the training dataset. We developed multiple models to test various parameter changes, and our results showcase the potential of machine learning to improve medical diagnoses.

However, it's worth noting that our dataset was small and limited to a single medical center in Guangzhou, China. Future studies could investigate the use of larger datasets and more sophisticated machine learning algorithms to enhance model accuracy and generalizability. Additionally, it is crucial to reduce the incidence of false negatives when using machine learning in the medical field. False negatives can result in missed diagnoses and negatively affect patient outcomes, so minimizing this risk is essential for ensuring model accuracy and reliability.

Despite these limitations, our project demonstrates the potential of machine learning to enhance pneumonia diagnosis through the analysis of chest X-ray images. By further refining and optimizing our models, they can be integrated into clinical practice to aid medical professionals in diagnosing and treating pneumonia. Our study highlights the significance of leveraging machine learning to improve medical diagnoses and ultimately improve patient outcomes.