Hearing Loss and Data Mining

Anil Chandler, Clemson University
Mathew Bernard, Clemson University
Mohammad Alshurbaji, Clemson University
Team 1

Abstract

Hearing loss is a very prevalent issue within society today. As the projection for hearing loss is projected to increase in the future, it is important to manage and predict this pattern for early detection and prevention. MRI scans are used frequently in the field of medicine and are able to accurately pinpoint issues within the brain region specific to hearing loss. By using specialized gray matter scans given by MRIs alongside using Data Mining techniques such as CNN, RNN, and LSTM models; finding thresholds will hopefully address this issue and will be able to benefit the general population in regards to the prevention of gradual hearing loss. By determining thresholds of hearing for individuals, possible prediction patterns could be used for those who might be at risk.

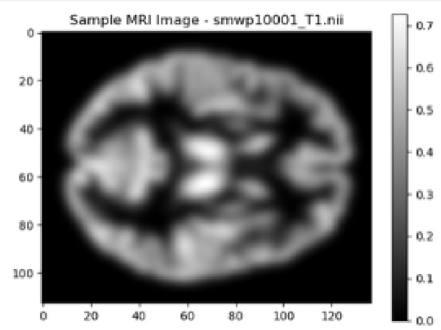

Figure 1: Gray matter scan of brain

Introduction

Hearing loss is described as being "not able to hear as well as someone with normal hearing, meaning hearing thresholds of 20 dB or better in both ears. It can be mild, moderate, moderately severe, severe or profound, and can affect one or both ears." [1] The abbreviation dB, better known as decibels, is measured based on how loud a sound is made. This loss of hearing can be caused by several factors including infections, loud noises, and aging. While each of these can be caused in synchrony with each other, aging has the primary potential to cause most major detrimental effects.

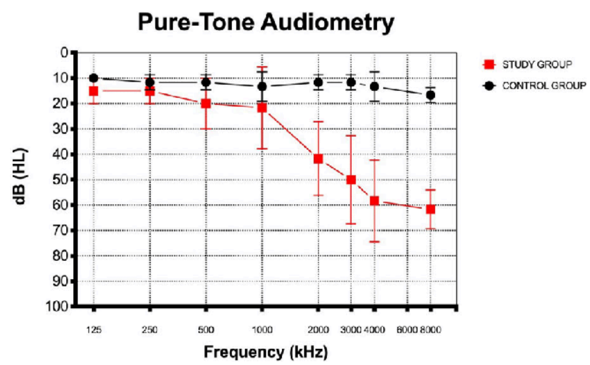

Figure 2: Pure-Tone Audiometry Test [2]

To measure the hearing ability of a certain individual, there is a tool called a pure-tone audiometer, which plays sounds on various frequencies and decibels to determine the ability of hearing. A test is usually performed in order to determine the ability of hearing. As shown in Figure 2, there is a graph depicting levels of decibels as well as frequencies shown in kHz which is also known as kilo-Hertz. A line is then drawn across what is able to be heard, which is determined to be the certain threshold to which someone could hear.

MRI (Magnetic Resonance Imaging) has been used prevalently ever since its inception in 1977. This discovery has quickly exploded making it a very prevalent and powerful tool. “To date, over 150 million patients have had MRI examinations. Every year, approximately 10 million patients undergo MRI procedures.” [3] From this increase in the technology, a certain type of special type of imaging stemmed from the MRI technology. This imaging type focuses on the gray matter within the brain. It is possible to see the correlation between a loss of gray matter to those with gradual hearing loss.

Given this information, it is possible to use data mining models and techniques to find these thresholds, and possibly predict what threshold would be considered to have hearing loss. This would be done using pre-made training information made in a dataset. Using the three models, CNN, RNN and LSTM, there is a possibility to accurately project whether or not a patient has hearing loss based on their MRI scans. There is also the potential to find the threshold of hearing loss based on MRI scans as well. This is what is planned to be accomplished over the duration of this research.

Methods

As is customary, among the critical initial phases of any data mining endeavor lies Data Cleaning and Data Preprocessing. In this research study, data preprocessing commenced with the extraction of hearing threshold data from an Excel file named "PTs_500_4k_blinded.xlsx". Initial steps involved filtering out negative values from designated columns such as "PT500" and "PT4000" to ensure data consistency. Subsequently, standard data cleaning procedures were employed to eliminate outliers and ensure data integrity. Utilizing the Z-Scores method with a threshold of Z=3, outliers were identified and removed. Consequently, the dataset was refined to 162 samples from the original 171, with 9 negative values and 1 outlier being excluded. These preprocessing steps were consistently applied across subsequent analyses.

ID	PT500	PT4000
smwp1_0001	25	73
smwp1_0002	3	10

Table1: Excel Data

Afterward, we proceeded to analyze the image data by visualizing them to gain insight into the nature of our dataset, followed by obtaining the dimensions of the 3D images. As illustrated below, the images have a 3D shape of [113, 137, 113], with intensity values ranging from a minimum of 0 to a maximum of 1.1. Normalizing the data to a range of [0, 1] was deemed crucial to facilitate smooth data processing. Subsequently, the decision was made to transform the image data from 3D to 2D, aiming to streamline our analysis and reduce computational burden, especially when handling large volumes of data. This approach is justified considering that the primary features of interest are concentrated within the first two dimensions.

Image 1	
Shape	(113,137,113)
Data Type	float64
Min Intensity	0.0
Max Intensity	0.859

Table2: Image Data

CNN

The Convolutional Neural Network (CNN) stands out as a premier deep learning framework for image analysis due to its tailored architecture, particularly designed to handle image data. Recognizing its aptitude, we selected CNN as our foundational model.

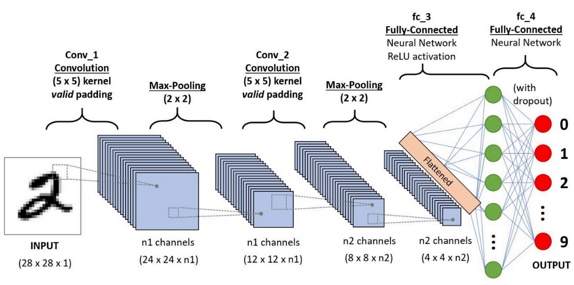

Figure 3: CNN Architecture [4]

In optimizing our data pipeline, we prioritized efficiency and computational economy by favoring 2D input data over 3D. This strategic choice not only streamlines processing time but also ensures robust performance even with large-scale data sets. Our approach involved reshaping the original 3D image data, transitioning from [113,137,113] to a 2D format [113,137,1]. While this may raise concerns about potential loss of depth information, it's important to note that our focus lies predominantly on the crucial features embedded within the first two dimensions, particularly evident in MRI scans.

Delving deeper into model refinement, we conducted a comparative analysis by implementing multiple variations. The initial model comprised 7 fundamental layers, including Conv2d layers, Max Pooling, Flattened layers, and fully connected layers. In contrast, our optimized iteration expanded to 15 layers, incorporating additional enhancements such as Batch Optimization alongside the core components of Conv2d layers, Max Pooling, Flattened layers, and fully connected layers.

RNN

Data loading for the RNN involved the retrieval of MRI data stored in NIfTI format from a tarfile ("n171_smwp1.tar.gz") and the modified preprocessed gray matter image data from a local directory set to ("/Users/…/8650dataset"), with the latter being directly loaded into memory for analysis. The dataset was then split into training and validation sets using an 80-20 split based on the IDs, with corresponding images paired with their respective hearing thresholds at 500Hz. Model construction ensued with the creation of a sequential neural network using TensorFlow and Keras, comprising three Simple RNN layers with 64 units each and a dense output layer with a single unit. Regularization techniques, specifically L2 regularization with a strength of 0.01, were applied to prevent overfitting. Following model compilation using the

Adam optimizer, mean squared error loss function, and mean absolute error metric, training commenced for 20 epochs with a batch size of 32, utilizing the training and validation data.

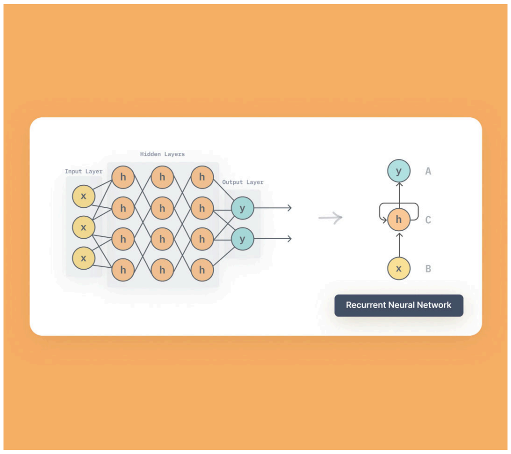

Figure 4: Recurrent Neural Network Architecture [5]

Evaluation was conducted on a separate test dataset, with reshaping of test data to match the input dimensions of the model. Both loss and mean absolute error metrics were calculated to assess the model's performance on the test data. Overall, these methods were pivotal in preprocessing the data, constructing the predictive model, and evaluating its performance, thereby contributing significantly to the research's objectives and findings.

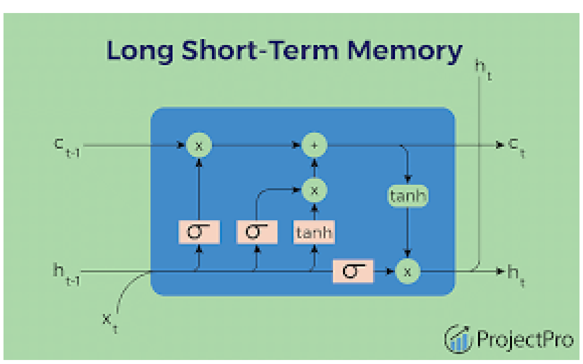

Figure 5: Long Short-Term Memory Diagram [6]

LSTM

LSTM is better known as Long Short-Term Memory, which is built in similarity to the RNN model. It is different in the sense that it fixes a vanishing gradient problem which arises during the weights and bias calculation and updating. LSTM uses leveraging to control and maintain strong gradients.

The dataset was split into training and validation sets similarly to the RNN model. A 80-20 split was used. The model uses TensorFlow and Keras which are programs used for tracking, monitoring, as well as retraining all information that comes in. The program has 1 Simple LSTM Layer (which uses 64 units), as well as 1 Dense Layer (set to 2 units). Just like the RNN, the model compilation also uses the Adam optimizer, calculates mean squared error loss, and calculates mean absolute error. The training for this model used 20 epochs with a batch size of 64.

Results

CNN

Both models yielded satisfactory results; however, as anticipated, the enhanced model outperformed its counterpart for several reasons:

- The enhanced model featured an increased number of layers.
- Epochs were extended from 20 to 50.
- Integration of the Batch Normalization Layer was implemented.

- Layer repetition was intensified, enhancing model depth and complexity.

Here are the achieved outcomes:

```
Mean Absolute Error: 13.248843743988961
Mean Squared Error: 279.55296847439433
```

Figure 6: CNN Model 1

```
Mean Absolute Error: 13.186280091603598
Mean Squared Error: 311.6369408432603
```

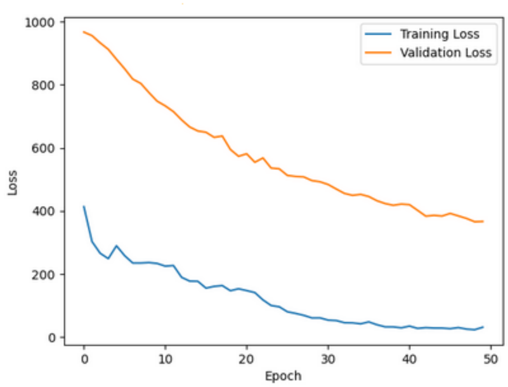

Figure 7: CNN Model 2

Based on these findings, it is evident that the CNN model demonstrates remarkable potential for predicting thresholds from MRI images, making it a highly recommended choice for industrial applications. However, optimizing its performance requires several enhancements, primarily expanding the dataset to encompass a broader range of data rather than a limited set.

Moreover, transitioning to Convolutional 3D architecture could further enhance performance, albeit with a slight increase in computational demands. Notably, adjusting the kernel size to [7,7] or [3,3] can significantly impact results. While larger kernel sizes offer greater precision, they also incur higher computational costs. Thus, a balance between precision and computational efficiency must be carefully considered when implementing these enhancements.

RNN

The output of the recurrent neural network (RNN) model indicates its performance and predictive capabilities following training and evaluation. Throughout the training process, the model exhibited a progressive reduction in both loss and mean absolute error (MAE), reflecting its ability to learn from the provided data. The final loss and MAE values on the validation set were recorded at 464.5618 and 17.3960, underscoring the model's ability to capture and generalize patterns within the training data. However, upon evaluation on an independent test dataset, the model displayed a slightly lower loss and MAE, with test metrics of 291.0109 and 16.2851, respectively. Although these metrics indicate a discrepancy between the model's predictions and the actual target values, they also highlight areas for potential improvement and further optimization. The test data shows that while there may be an error in the model it is consistent when tested against unseen data. Overall, the completion of the model build and evaluation stages provides valuable insights into its performance and future refinement.

LSTM

The output of the Long Short-Term Memory model (LSTM) shows its ability to evaluate and integrate the incoming information. There are four main metrics looked at for the results of this model. These metrics are

Validation Loss, Test Loss, MAE (Mean Absolute Error), and MSE (Mean Squared Error).

Validation Loss is the difference between predicted and target output. Test Loss is the estimate of how well the model trains new incoming data. MAE is the measure of error in the average variance between significant values. MSE is the average squared difference between the observed and predicted values.The goal of these values is to have each of them as close to zero as possible.

For the model, the Validation Loss was calculated to be 979.2110. The Test Loss was calculated to be 689.1660. The MAE was recorded to be 18.4779. The MSE was recorded to be 689.1660. These numbers are considered to be much higher than the target values that the project was aiming for.

Conclusion

Research shows that Convolutional neural networks (CNNs) excel at capturing spatial patterns, like those found in images. Meanwhile, simple recurrent neural networks (RNNs) and long short-term memory networks (LSTMs) are good at capturing temporal dependencies over time. Where temporal dependencies refer to the relationships between data points at different time steps. For example, the price of a stock today may be influenced by its price yesterday, the day before, and so on. In the study we have shared today the brain scan images aren't dependent on the previous which might give insight to why the RNN and LSTM models performed worse than the CNN. And our findings confirm the research and based on our findings we can conclude that CNN performed the best, then simple RNN and lastly LSTM performed the worst.

Works Cited

[1] - World Health Organization: WHO. "Deafness and Hearing Loss." *Who.int*, World Health Organization: WHO, 18 Sept. 2019, www.who.int/health-topics/hearing-loss#tab=tab_1.

[2] - "Figure 1. Results of Pure Tone Audiometry (PTA) Test in Healthy..." *ResearchGate*, www.researchgate.net/figure/Results-of-pure-tone-audiometry-PTA-test-in-healthy-controls-and-MD-subjects_fig1_339140282.

[3] - Admin. "Information for Patients." *ISMRM*, www.ismrm.org/resources/information-for-patients/#:~:text=Doctors%20use%20MRI%20to%20examine.

[4] - "Papers with Code - an Overview of Convolutional Neural Networks." *Paperswithcode.com*, paperswithcode.com/methods/category/convolutional-neural-networks.

[5] Bagheti, Pragati. "The Complete Guide to Recurrent Neural Networks." V7, www.v7labs.com/blog/recurrent-neural-networks-guide. Accessed 19 Apr. 2024.

[6] - "The Ultimate Guide to Building Your Own LSTM Models." *ProjectPro*, www.projectpro.io/article/lstm-model/832.