Name: Jason Zhou
Mentor: Dr. Dongjin Song
Status Report #: 7
Time Spent on Research This Week: 6.5
Cumulative Time Spent on Research: 39.5
Miles Traveled to/from Mentor This Week: 0
Cumulative Miles Traveled to/from Mentor: 0
==========

Monday, October 25th, 2021: (3.5 Hours)

To start this day, I had my weekly meeting with my mentor. Usually, the meetings are on Friday; however, my mentor wanted to move it to Monday from now on because he will now be busy later in the week. During our Zoom meeting, I explained to him that I was getting my articles through the DCASE 2020 website and was simply taking papers that were based on a certain machine learning model. He told me what I was doing was fine and that I should not stress about going too in-depth because he really only had a couple of models that he wanted me to research when he proposed this project to me (i.e classical methods and some deep learning methods).

After finishing the meeting, I entered the final stretch of my literature search. I started with the article IAEO3 - COMBINING OPENL3 EMBEDDINGS AND INTERPOLATION AUTOENCODER FOR ANOMALOUS SOUND DETECTION. They propose two different systems for handling this problem: a general system for all machines and a system with hyperparameters specific to the machine. Both systems are split into two branches. In the first two branches, audio embeddings[1] are extracted from the data and are normalized. Afterward, principal component analysis[2] is used to reduce the size of the data and create a feature vector. In the second branch, STFT is used to extract 5 mel-spectrograms from the data. The spectrograms are lined up and the middle one is taken out. Then, an interpolation autoencoder constructs a new spectrogram to replace the middle one that was just removed. Both the new and old framed are compared to calculate a reconstruction error. Based on this reconstruction error, statistics like mean, median, and standard deviation are recorded and combined with the feature vector that was previously made. These are imputed into a GMM and an anomaly score is calculated.

Next, I dove into a paper titled An Ensemble Approach to Unsupervised Anomalous Sound Detection. In this paper, they use two different methods: GMMs in combination with fisher vectors[3] and a copy-based algorithm. 6 fisher vectors are created (one for each machine type). Then, these vectors are inputted into a GMM to calculate the anomaly score. In another approach, the authors employ a copy-based algorithm. They take the first 20 audio samples and

[1] Low-dimensional vectors that contain audio features
[2] An algorithm used for dimensionality-reduction. Essentially, the algorithm takes the data and trashes anything that is not useful, making the data more compact.
[3] Fisher vectors are just ways of storing features, similar to i-vectors or embeddings

compare them with the rest to find points of similarity. They use these to create templates [4]of what is considered normal sounds. Then the templates are used in the evaluation data. The algorithm calculates the difference between the template and the evaluation data. The higher the difference, the higher the likelihood that the evaluation data is an anomaly.

I then began reading REFRAMING UNSUPERVISED MACHINE CONDITION MONITORING AS A SUPERVISED CLASSIFICATION TASK WITH OUTLIER-EXPOSED CLASSIFIERS, which uses a different theory for classification and a ResNet model for anomaly detection. This article attempts to redefine how a model will train with sounds. Specifically, it considers outliers to be considered as anomalies. Outliers are usually anomalies; however, this is not always the case. In some cases, outliers can be neither normal nor abnormal. By classifying outliers as anomalies; however, the model can train off of "anomaly" data. This is a confusing concept, so I am not sure if I have fully understood it. I will have to look into this further. Besides this process, though, similar to other articles, the data is passed into the machine learning algorithm.

Afterward, I read a paper called Acoustic Anomaly Detection for Machine Sounds based on Image Transfer Learning. This paper was fairly unique in its approach compared to other papers that I have read. Just like other papers, these researchers utilized a mel-spectrum gram to visualize the audio file. However, instead of extracting sound characteristics/features from the mel-spectrogram, they simply treated it as an image. By treating it as an image instead of a mechanism to obtain sound data statistics (frequency, power, volume, etc), they found that normal sound has a common appearance and "looks" the same. By applying image recognition algorithms, they can train a model to learn the difference between normal and abnormal sounds.

I then moved on to an article named ANOMALOUS SOUND DETECTION BASED ON INTERPOLATION DEEP NEURAL NETWORK. This uses an interpolation deep neural network (IDNN). Essentially, they took an audio clip and split it into five parts. From these five parts, five mel-spectrograms were created and lined up in chronological order. Then, they remove the middle mel-spectrogram and have the IDNN interpolate the middle part, hence the name. This apparently makes the reconstruction error more precise and consistent across different data. Ultimately, this led to massive improvements in the AUC. Specifically, they say that it was 27& improvement compared to the standard/baseline.

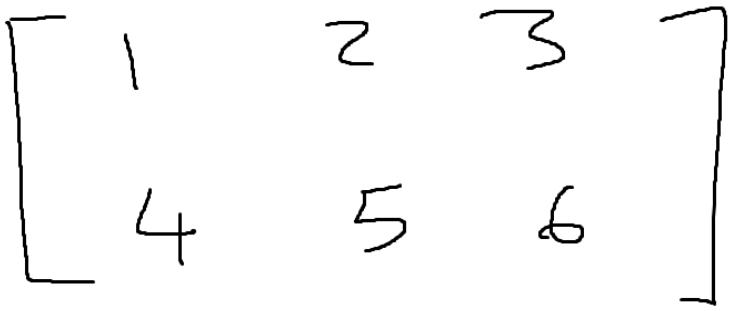
Additionally, I read a paper titled ANOMALY DETECTION BASED ON AN ENSEMBLE OF DEREVERBERATION AND ANOMALOUS SOUND EXTRACTION. As the name suggests, this paper focuses on the dereverberation of sound. This is basically the removal of background noise or reverberations within the audio file. Thus, by removing the noise, the model is better able to predict anomalies. I suspect this is because the model is no longer distracted by the influence of noise and can, therefore, make more accurate guesses. Then, they place their denoised data into an autoencoder and get an anomaly score.

Wednesday, October 27th, 2021: (1 Hour)

[4] I assume these templates contain features that are characteristic of a normal sound.

While reading the academic articles for step one of the research proposal, I had a hard time understanding what was being talked about. Many of the articles involved complex math equations and terminology that I believe was from linear algebra, which I do not know. As such, I decided it would be important to at least learn the basics of linear algebra before diving further into my research project.

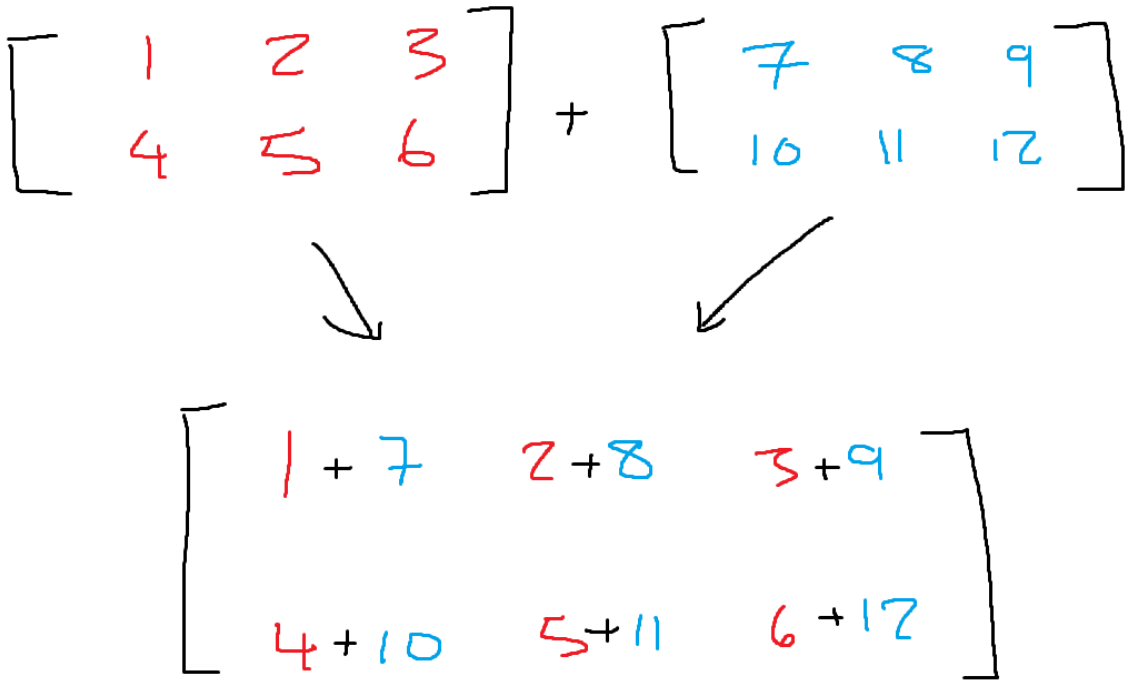
I quickly looked up a video on youtube about linear algebra, which went over various concepts that were important for machine learning. Although I had already learned some of the concepts from PreCalculus (such as vectors), I learned a variety of new concepts such as what a matrix was.



(This is a picture of a matrix, which is essentially just a 2D array of numbers)

At the simplest level, a matrix is just a collection of numbers(elements) that are ordered into rows and columns. In the above picture, one would call this a 2x3 matrix because it has 2 rows and 3 columns. In the video I watched, I learned how to perform different mathematical operations on matrices such as addition, subtraction, and multiplication. However, in order to do such operations, certain conditions have to be met. For example, when adding or subtracting matrices, they both have to be the exact same shape (size). This is because each number in a matrix is added/subtracted with its “partner” in the other matrix. By adding or subtracting matrices, creates a new matrix with the combined values; however, the new matrix must have

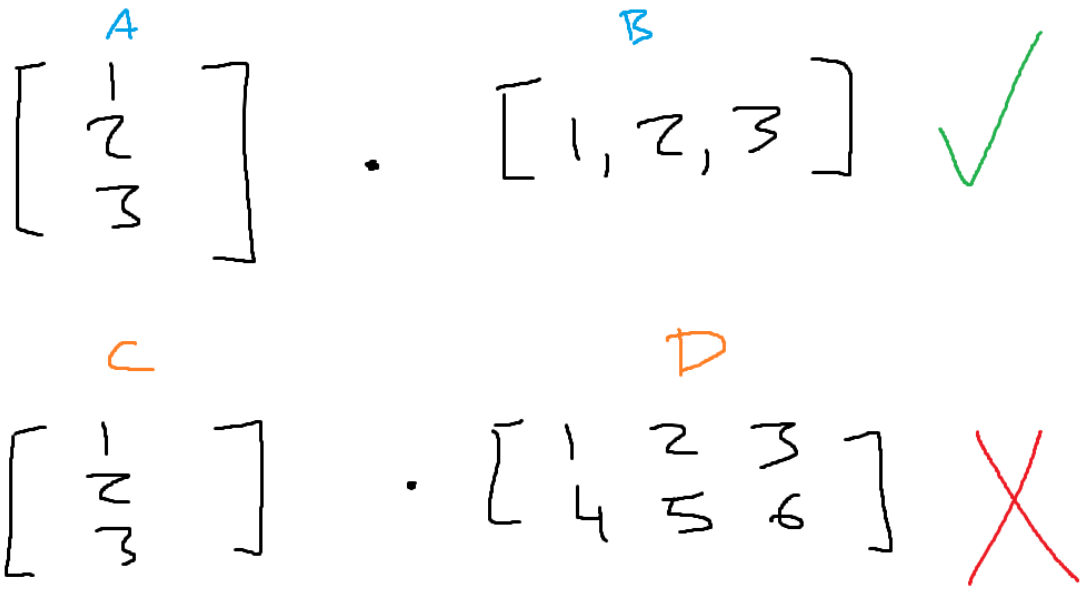
the same number of elements in it as the previous 2.



(This is a picture of matrix addition between two matrices. Notice that the elements (numbers) from each matrix have been color-coded to better display what is happening. Each number from one matrix has to be added/subtracted to another from the other matrix. If a number has no “partner” a math operation cannot occur between the two matrices)

Matrix multiplication is a little bit different. Instead of requiring both matrices to have the same number of elements, multiplication only requires that one matrix has the same number of

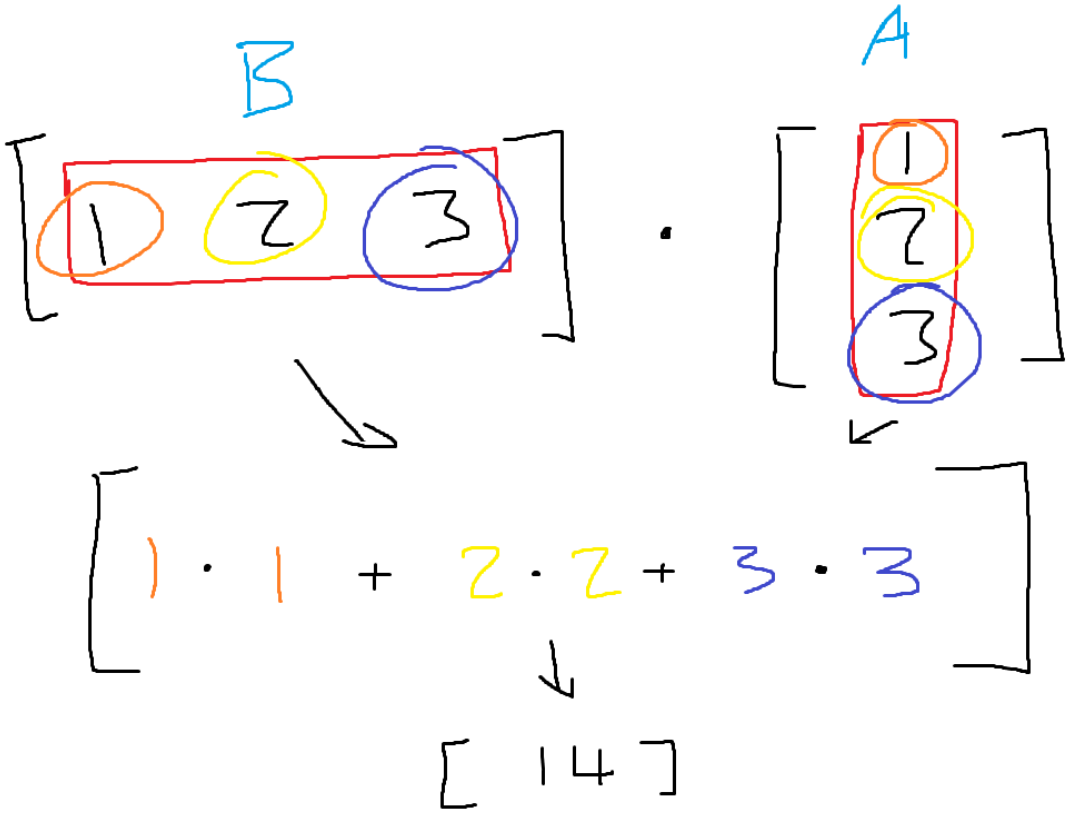
columns as the other matrix has rows.



(This is a picture that shows valid and invalid examples of matrix multiplication. Notice that each matrix has been labeled with a letter to make referencing it easier.)

For example, Matrices A has one column, while Matrix B has one row. Because Matrix A has an equal number of columns as Matrix B has rows, they are able to be multiplied. To actually multiply together, one needs to multiply the rows of Matrix A with the columns of Matrix B. Within each multiplication of a row and column, the individual elements are multiplied. Specifically, in the picture below, an element in the first spot on the row of Matrix B would be multiplied by the first element in the column of Matrix A. This would repeat for the second element in both Matrices, and then the third, and so on. As long as one is multiplying the nth row by the nth column, one must multiply all “partnered” numbers and add the products of those

with the rest of the products that resulted from multiplying the row and column.



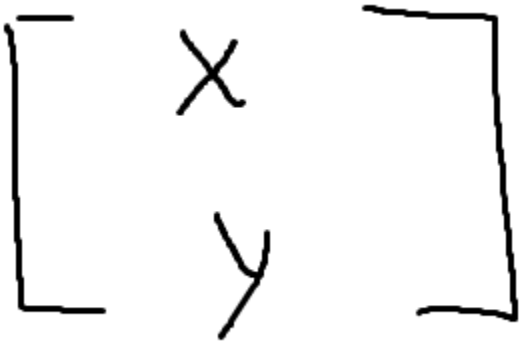
(An example of matrix multiplication, which results in a single element (14) within a matrix. In reality, A and B are just 3D vectors; however, the operations for vectors and matrices are the same in this case. Thus, I refer to A and B as matrices because it makes the explanation easier to understand)

One other to note is that matrix multiplication is not communicative like regular multiplication is. For example, if one were to multiply matrix A by matrix B to get matrix AB, this would be different than multiplying matrix B by matrix A and getting matrix BA. Simply put, matrix AB does not equal matrix BA (would not have the same elements).

Friday, October 29th, 2021: (0.5 Hours)

On this day, I continued to learn linear algebra. The video watched talked about vectors, which I had learned in PreCalc, but included some interesting terminology that had come up in many of my articles.

Essentially, vectors can be expressed within brackets like this:



(A picture of mathematically representing vectors within brackets)

This represents one way that vectors are expressed on a coordinate plane. If the beginning of the vector begins on the origin, then x would be the vector's x-component, while y would be the vector's y-component. Additionally, vectors are not limited to just x and y. They can have many more components such as a z-component, which would make the vector 3D. In general, a vector is in the nth dimension based on how many components it has.

In my articles, vectors were often used and accompanied by a strange notation that I could not quite understand. After learning more about vectors and how they can be expressed, I learned this notation:

(A picture displaying a common notation I saw throughout my literature search)

The V represents a vector. The ∈ means "exist," while R^n simply means the nth dimension. Although I was baffled by what this meant, it now makes sense to me. Essentially, one could read this notation as "vector V exists within the nth dimension." So if n was equal to 5, then vector v would exist within the 5th dimension, which just means that v is a 5D vector (5 components).

Sunday, October 31st, 2021: (1.5 Hours)

After reviewing what I had learned this week (matrix operations), I decided to actually get some practice in with machine learning. I decided to try and code an autoencoder, which was a machine-learning algorithm that had come up frequently in my literature search.

The basic principle of an autoencoder is fairly simple. It takes in an input, deconstructs it, and then reconstructs it. In this way, the algorithm basically outputs what was inputted into it. However, during the process of deconstruction, the algorithm throws out most of the data and only keeps the most important ones. Using the leftover data, it then reconstructs the data to the best of its abilities. This would be akin to taking a puzzle, throwing out most of the pieces, attempting to reconstruct the missing pieces, and trying to put it all back together. As one can imagine, the reconstructed, or puzzle, probably will not look exactly the same, which is precisely why autoencoders are useful in anomalous sound detection. Autoencoders usually output data based on the sample sets that were used to train them. Thus, if an autoencoder finds a piece of data that is not closely related to the sample set, it will reconstruct it completely wrong. In other words, it rebuilds the puzzle wrong. Thus, one can train an autoencoder purely on audio data that is considered normal for an environment. Then, when an autoencoder encounters an anomalous sound, it will typically reconstruct the sound poorly, making it easy to spot the anomalous sounds within data.

I watched a video to help me with the process of making an autoencoder. First, I created an encoder (a model which compresses the data, thereby throwing out most of the data), and then I created decoding later (a model which decompresses the data, allowing for the reconstruction of data). This was essentially the entire process.

Additionally, the video I was watching used functional API, which is a more advanced way of building machine learning algorithms than I had previously been using (I had been using what is known as sequential API). This is what took up most of my time when learning how to make an autoencoder. Functional API allows one to make more complicated models by allowing the user to freely connect and disconnect layers with one another. Although the concept of this API sounds simple, the video I watched did not explain it very well, so I had to do some digging and figure out exactly what was going on.

References

3Blue1Brown. (2016, August 5). *Vectors | chapter 1, essence of linear algebra* [Video]. YouTube. https://www.youtube.com/watch?v=fNk_zzaMoSs

The functional API. (n.d.). TensorFlow. Retrieved November 1, 2021, from https://www.tensorflow.org/guide/keras/functional

Geek's Lesson. (2019, August 14). *Linear algebra for beginners | linear algebra for machine learning* [Video]. YouTube. https://www.youtube.com/watch?v=kZwSqZuBMGg&t=3218s

Jaadi, Z. (2021, September 16). *A step-by-step explanation of principal component analysis (PCA)*. Builtin. Retrieved November 1, 2021, from https://builtin.com/data-science/step-step-explanation-principal-component-analysis

Khan Academy. (2009, October 15). *Matrix vector products | vectors and spaces | linear algebra | khan academy* [Video]. YouTube. https://www.youtube.com/watch?v=7Mo4S2wyMg4

The Organic Chemistry Tutor. (2018, February 17). *Multiplying matrices* [Video]. YouTube. https://www.youtube.com/watch?v=vzt9c7iWPxs

sentdex. (2021, March 1). *Autoencoders in python with tensorflow/keras* [Video]. YouTube. https://www.youtube.com/watch?v=JoR5HCs0n0s

Grollmisch, S., Johnson, D., AbeBer, J., & Luka-shevich, H. (2020). IAEO3-combining OpenL3 embeddings and interpolation autoencoder for anomalous sound detection. *Tech. Rep., DCASE2020 Challenge*.

Alam, J., Boulianne, G., Gupta, V., & Fathan, A. *An Ensemble Approach to Unsupervised Anomalous Sound Detection*. Technical report, DCASE2020 Challenge (July 2020).

Primus, P. (2020). Reframing unsupervised machine condition monitoring as a supervised classification task with outlier-exposed classifiers. *Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE 2020 Challenge), Tech. Rep*.

Müller, R., Ritz, F., Illium, S., & Linnhoff-Popien, C. (2020). Acoustic anomaly detection for machine sounds based on image transfer learning. *arXiv preprint arXiv:2006.03429*.

Suefusa, K., Nishida, T., Purohit, H., Tanabe, R., Endo, T., & Kawaguchi, Y. (2020, May). Anomalous sound detection based on interpolation deep neural network. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 271-275). IEEE.

Kawaguchi, Y., Tanabe, R., Endo, T., Ichige, K., & Hamada, K. (2019, May). Anomaly detection based on an ensemble of dereverberation and anomalous sound extraction. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 865-869). IEEE.