CS 753: ASR
Project Report: Explainability of ASR

Nikhil Chavanke:170050017
Siddhesh Pawar: 17D170011

Introduction

In this project, we probe the attention hidden layers of transformer-based hidden layers to understand acoustic processing that occurs across hidden layers in an end-to-end ASR model. We specifically probe facebook's wave2vec2.0 model to get the hidden layer activations and then use the activations to train simple classifiers to perform various downstream tasks such as speaker classification, phone classification, and word classification. (we initially had started with Mockingjay and Audio ALBERT following this paper[1], however, Mockingjay's code was giving errors in preprocessing of some datasets, which we couldn't resolve, and later's code was not available so we shifted to wav2vec2.0).
We test the hypothesis that the deeper the hidden layers(with shallowest layers defined as the first transformer layer from the input) learn more complex acoustic and language phenomena, we also quantify which hidden layer of the model, learn which phenomena through experiments. We finally provide a Jason Shannon Divergence perspective to the extent of learning that occurs across various layers, in order to quantify the similarities of attention distribution for similar acoustic features. We also carry out extensive visualization through t-SNE to validate the results that we got through the training of downstream classifiers.

Literature Survey

We started our project by reading the paper 'Analyzing Hidden Representations in End-to-End Automatic Speech Recognition Systems' by Belinkov and Glass[2], parallelly we also read the Ph.D. thesis of one of the authors, Belinkow which was entirely based on language representations learned by machine Translation and ASR systems, the thesis also gave a thorough an introduction to explainability techniques. We initially thought of reproducing it but had to cancel our plans as the entire codebase was in Lua programming language. We then read the papers, 'How Accents Confound: Probing for Accent Information in End-to-End Speech Recognition Systems'[3] and 'What all do audio transformer models hear? Probing Acoustic Representations for Language Delivery and its Structure'[4] to converge on the task of probing for accents, phones of various models as the major task. We started this task by searching for models that could be probed and looked at AALBERT [1] paper initially and decided to use techniques used in the paper. The paper also introduced us to the concept of JS divergence as a way of quantifying the amount of learning between the attention layer, which we further explored by reading 'Adaptively Sparse Transformers' [5] and the classic 'Divergence Measures Based on the Shannon Entropy'[6] by Jianhua Lin which introduces JS divergence as a tool to measure similarity between random graphs (the attention weights can be considered a random graph). We also read the audioLIME [7] paper, which we thought was the ASR equivalent of the popular technique LIME used in NLP, however, the code was too complicated to reproduce and the code required medium and large music clips, so we didn't pursue that direction due to lack of computing power.

Datasets and getting around the large data size

We initially started with using the LibriSpeech dataset, but soon our colab sessions started crashing due to the size of the dataset (and also size per sample was large). So we shifted to lighter datasets such as TIMIT which we used for speaker classification and phone classification. We used the CMU - ARCTIC dataset which had data for 18 different speakers(from different regions and speaking different dialects) to probe for accent. We also used the speech command dataset to probe for word-level features. For, all the above datasets, the output of the model was of size (Number of hidden layers)*(sequence length)*768, and the attention weight matrix was of size (Number of hidden layers)*(Number of heads)*(sequence length)*(sequence length) which could not be stored for all the data point on colab, all at once so we separately pickled the hidden activation and attention weights iteratively for each data point, and then used the pickle files on the local machine for experiments

Experiments and Discussion

The model that we used was the base wav2vec2.0 which was retrained and fine-tuned on 960 hours of Librispeech on 16kHz sampled speech audio.

1. The First task was the one of accent classification, here firstly we trained a binary classifier to classify male and female voice having:
    a. Same accent (Midwestern accent data from CMU dataset was used)
    b. Having a different accent (Midwestern vs Scottish accent)

As we can see in figure 1 below, the classifier is actually able to classify well for the first few hidden layers but starts getting confused after layer 4 for the same accent and layer 7 for different accents. This is due to the fact that the initial layers learn low-level features such as tone, modulation and as they are different for different genders, the classifier easily classifies them. The higher layers learn more complex features such as pronunciations and as the high-level features are the same across accent, the classifier starts to get confused(accuracy falls), whereas in the different accent case, even the pronunciation features are different, so the classifier performs well when the accent is different. The eventual drop in the accuracy can be explained by the fact that at the end the speech is converted to text and written text often remains common across various accents and the deeper layers actually earn the text(and thus confuse the classifier). The same can be seen when training a classifier to distinguish between two related dialects vs two different dialects. As can be seen in Figure 2, the second plot represents a classifier trained to classify two similar accents Canadian(jmk) and American(bdl)

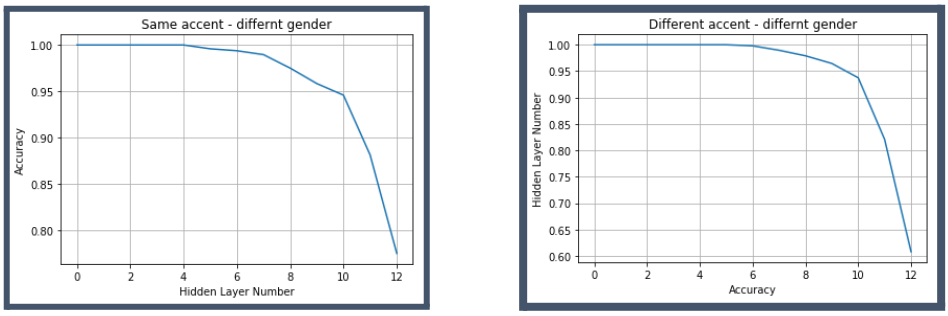


Figure 1: Gender Classification

(Note that jmk and bdl are just names of the speakers)

And thus gets confused from the 6th layer while in the first plot, it is trained to classify Scottish vs Canadian which is strikingly different from each other as pointed out in this report. We,

support this claim by comparing **JS divergence** pattern between the hidden layers of wav2vec2.0 model

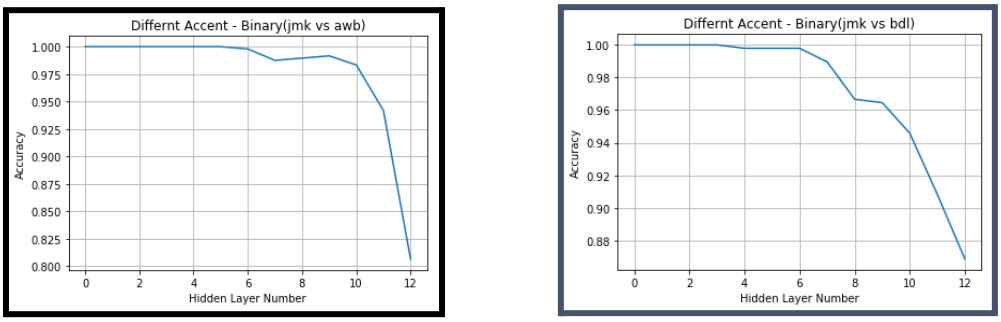
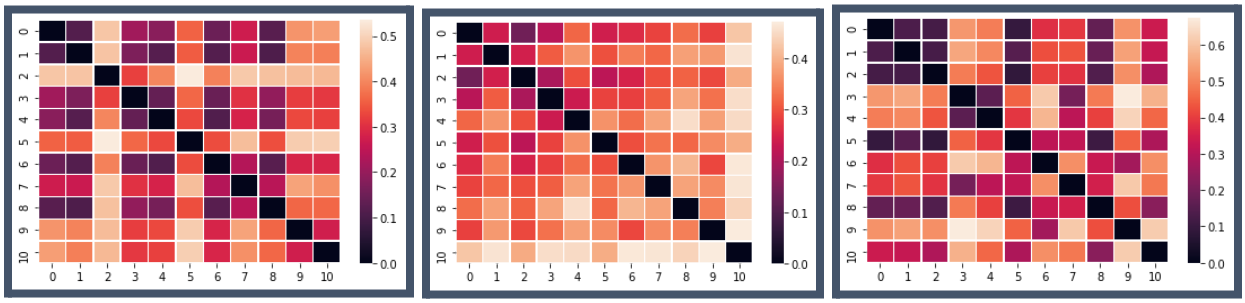


Figure 2: Related vs unrelated dialects

The JS divergence measures disagreement between the attention layers, the higher the value, the more the layers are disagreeing with each other in terms of where to attend, and thus higher the value of disagreement between layers implies that those two layers are learning different phenomena, whereas lower value implies redundancy. This can be seen in figure 2, the pattern is the same for JMK(Canadian) and BDL(American) whereas it is different for the speaker AWB(Scottish), the similarity between these patterns was calculated by taking cosine similarity between corresponding rows and then taking an average and it comes out to be 0.003 between AWB and JMK while it is between JMK and BDL it is 0.006 (note that the value of JSD is between 0 and 1, and so the magnitude of dot products is less, so these values should be interpreted as relative numbers rather than absolute similarity values). Also **note** that each block (i,j)in the grid represents JS divergence between attention weights of ith and jth hidden layer



Speaker: JMK Speaker: AWB Speaker: BDL

Figure 3: JS Divergence between accents

Now we train a multiclass classifier, for 12 senses, and the accuracy as can be seen in the first part of Figure 4 is almost 1 for the initial 2-3 layers. This is due to the fact that the actual processing of speech features is done by CNNs which are before the transformer layer in the wav2vec2 model(Note we feed raw sound signals to the model), this can also be verified by looking at the JS divergence between adjacent layers as shown in the first part of figure 5.

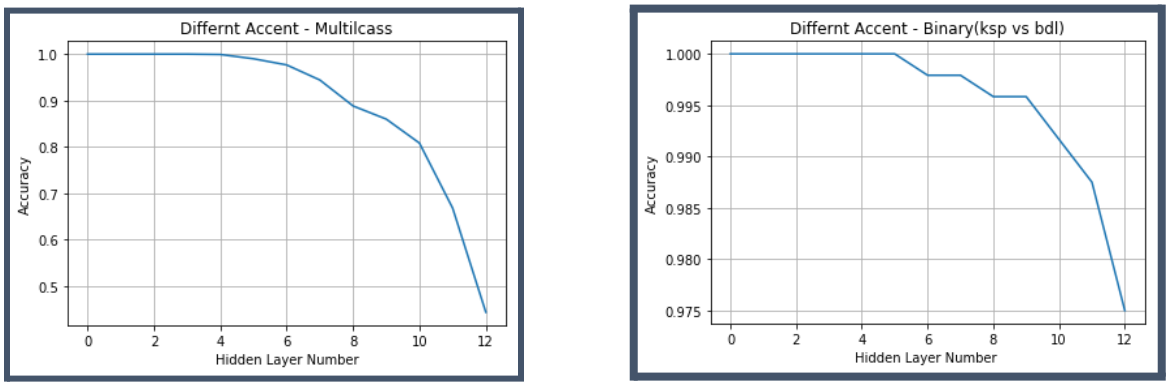
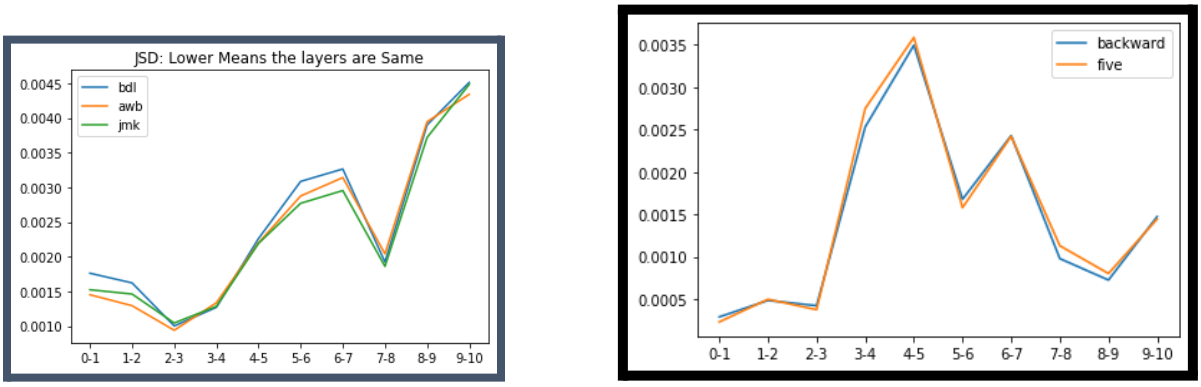


Figure 4: Multiclass classification

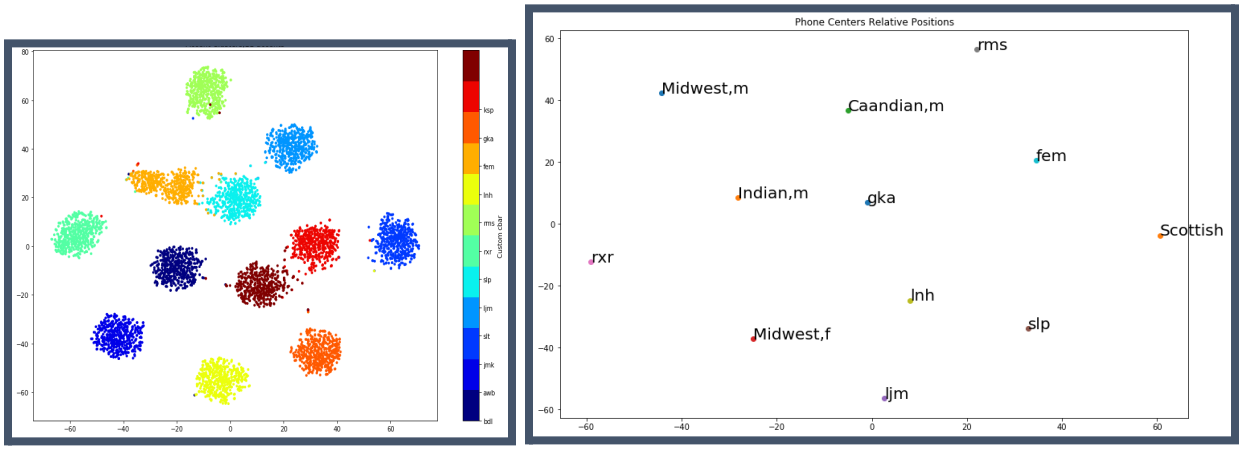
For the initial 2-3 layers, the JS divergence is low implying that almost no learning is taking place and the features extracted by CNNs are directly passed through these layers, thus implying high accuracy across these layers. These classes have been plotted using t-SNE for



JSD for various accent JSD for words ‘backward’ and ‘five’

Figure 5: JS divergence between adjacent layers

layer 2, we can see that these clusters are neatly separated out. One interesting observation is that, note that clusters of Scottish and Midwest accent lie far apart while that of midwest and Indian are close, to each other this can be also verified by the second plot in figure 4 which implies that Indian and Midwestern accents are actually confusing for the classifier from the 4th layer (and hence are similar). Thus we can see that CNNs are strong processors of speech signals while the transformers are strong processors of interdependencies between the signals in a sequence. Also, note that the deeper layers of transformers process the English language



Clusters Cluster centroids

Figure 6: t-SNE plots for accent (jnh,slp, etc are names of speakers)

which is independent of the accent. As we can see in Figure 7, the accent information is lost or filtered out in the deeper layers, this also explains the fact that the representations of deeper hidden layers end up confusing the accent classifier.

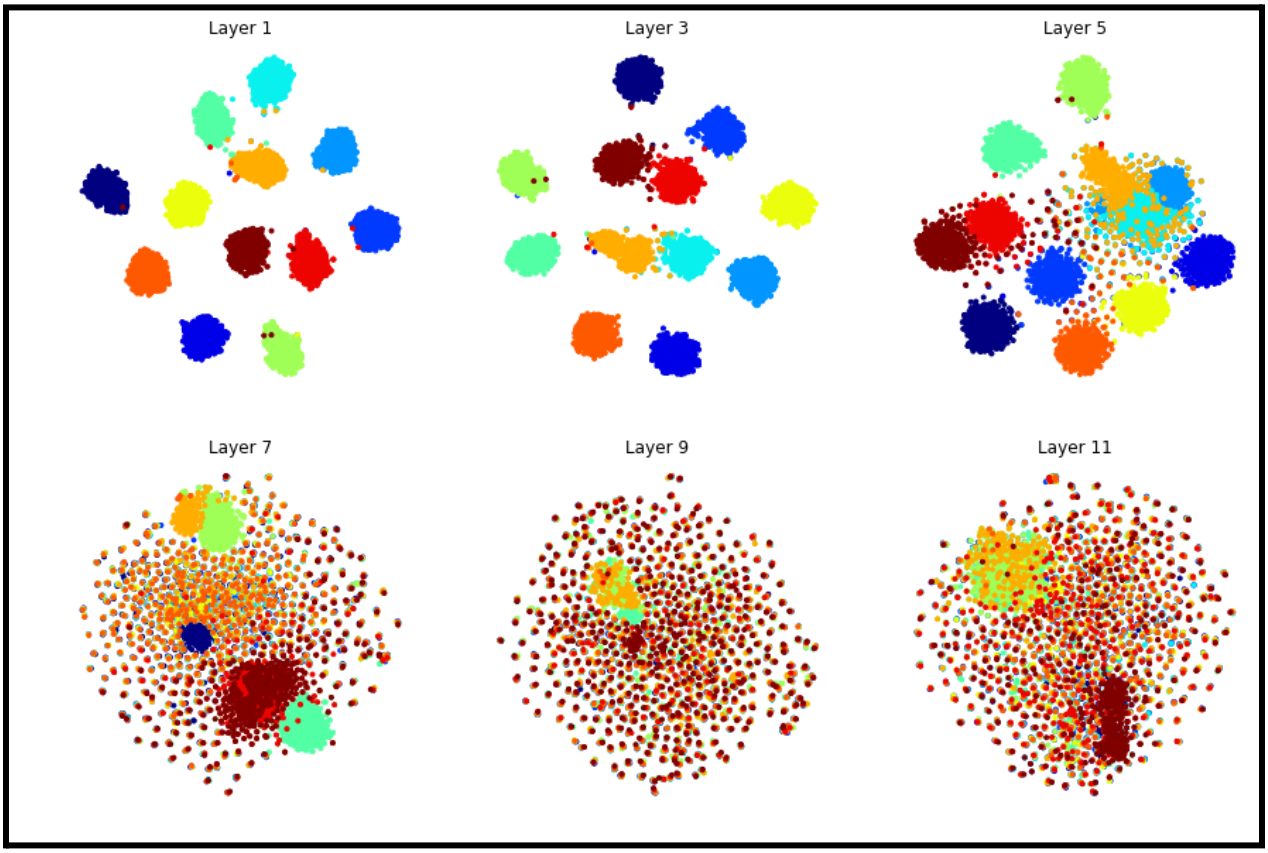


Figure 7: Accent clusters across various hidden representations

3. **Speaker Identification** :

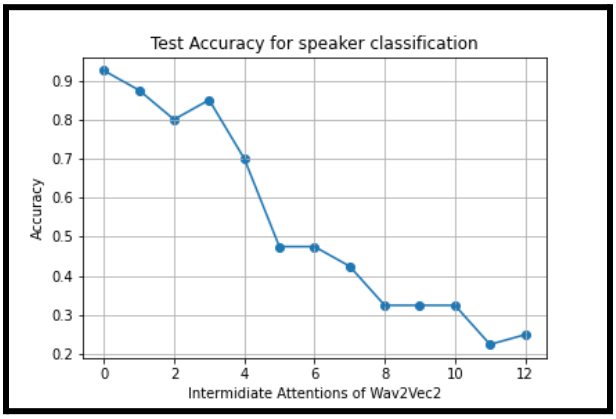


Figure 8: Speaker Identification

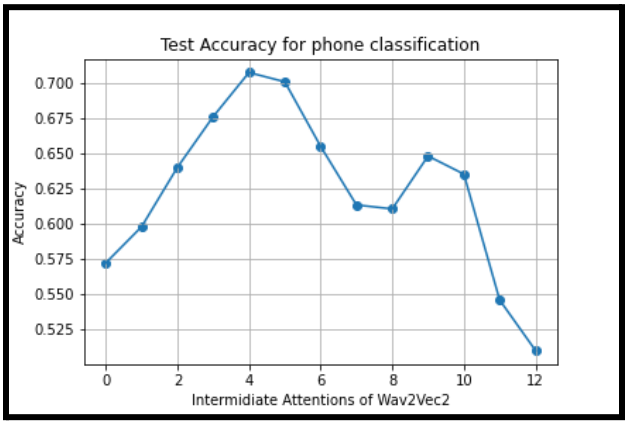


Figure 9: Phone Classification

In this task, we performed a multiclass classification problem for Speaker Identification. The dataset chosen for this purpose was TIMIT, which includes around 600 speakers. But for the task, only data corresponding to 25 speakers was taken into consideration. So overall speech utterances were used for the purpose. Representations from the 13 attention layers of wav2vec2.0 were used as features. After getting the representations, the output is averaged pooled across the time frame dimension, the reason being the speaker features should be independent of the timeframes. Now the 768 size representations are passed on to Support Vector Classifiers, each for the 13 layers. The plotted graphs show the test accuracy over the layers. In Figure 8 we can see that we get maximum accuracy using the first layer representations. This is very intuitive as the model is expected to be speaker-independent and hence along with the layers, speaker information is lost progressively.

4. **Phone Classification:** In this experiment we perform phone classification, again using the intermediate attention layers of wav2vec as the feature extractors. For this purpose, the TIMIT phonetic dataset is used. This includes a total of 61 phone classes in the English language. As the dataset consists of sentences, we need to modify the data to be able to perform phone classification. This data actually also provides the start timeframe and stop time frame for each phone in a particular sentence. So what we do is divide the total speech frames in an utterance into groups such that they correspond to a particular phone. And each speech frame group and its phone label are used as a single data instance. For this task, the classification is done similar to the Speaker identification task using attention. And we can see in Figure 9, the phone representations are best learned by the 4th layer.

Now we also use PCA followed by t-SNE for visualization of the representations. So dimensionality reduction is applied for the 4th layer and the centroids are plotted. This is done by first reducing the dimension of the representation from 768 to 50 using PCA and then using t-SNE to reduce it further to 2 dimensions. In Figure 10, the class cluster centroids are plotted. Here as we can see all the vowel phones are adjacently represented by blue color. The ones in the red are all the Nasal phones, while all the phones highlighted with green are voiceless phones(voiceless sounds). All phones other than the blue highlighted use mostly consonants.

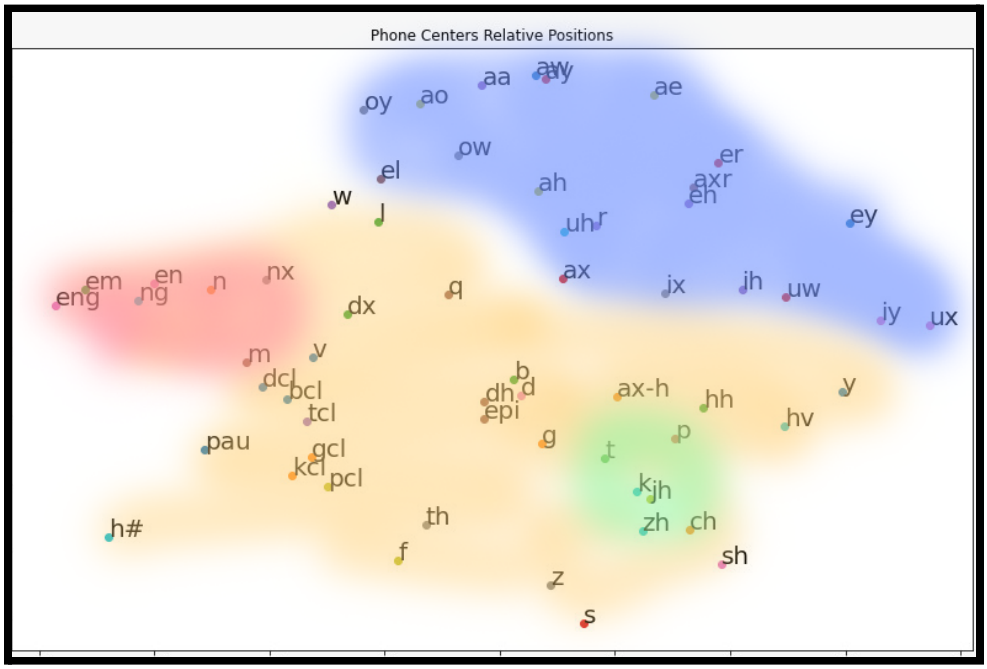


Figure 10: Centroids of phone clusters in t-SNE 2d representation

4. **Speech Command Classification(Word Classification):** In this task, we consider a word classification problem. The dataset used for the purpose is Speech Commands Dataset. This dataset includes 35 speech commands each of which is a single word. Same way as phone and speaker here we use SVCs for the task. In Figure 11, we can observe that the peak of the graph is at attention layer 7. This makes sense as words are more complex structures than phones. And as phones were learned in the 4th layer, in a couple of more layers, it combines the phone information and derives word representations.

Here also we use the visualization technique of t-SNE, to get the plot in Figure 12. In this plot, similar sounding words are found to be adjacent to each other. Like “house”, “wow”, “down” are close to each other. From this, we can infer that in between layers 4-7 the model is learning combinations of phones, as up to the 4th layer it has already learned phones. This might be the reason that in Figure 9 we see the trough in between layers 4-7. As it might be focusing on the sequencing of the phones rather than learning more acoustic features. After layer 6-7 we see a rise again.

Now for the Speech commands, we plot the 2-d representations over layers 3,5,7,9,11,13 and get the plot in Figure 13. Initially, there are no separate class clusters. Then in layer 7 we get the best distinct clustering. After that, the model retains information of distinction of certain words but some clusters are mixed again. These clusters might be the ones corresponding to the short words like “wow”, “two”, “one”, etc. As long term speech information is captured in layers after attention layer 7.

This fact can also be verified from the second plot of figure 5, wherein the JS divergence patterns are almost the same for most of the initial layers, and the JS divergence value is different for the 7th and 8th layer, implying the learning patterns(attention weights) are different for those layers.

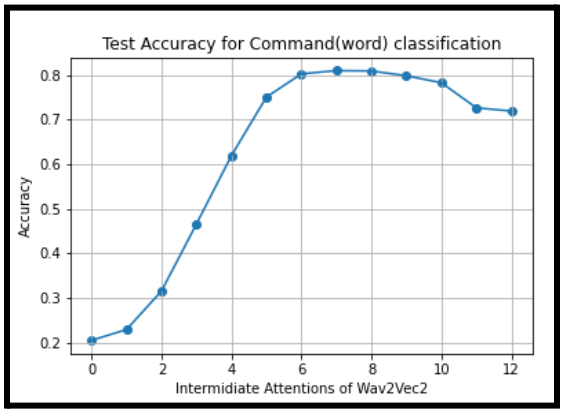


Figure 11: Word Classification

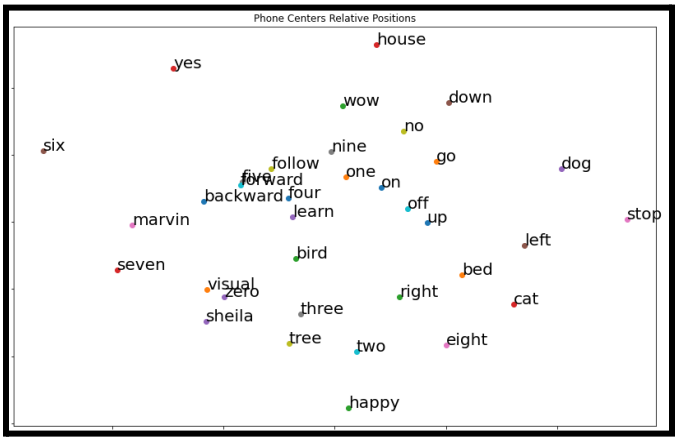


Figure 12: Centroids of t-SNE clusters

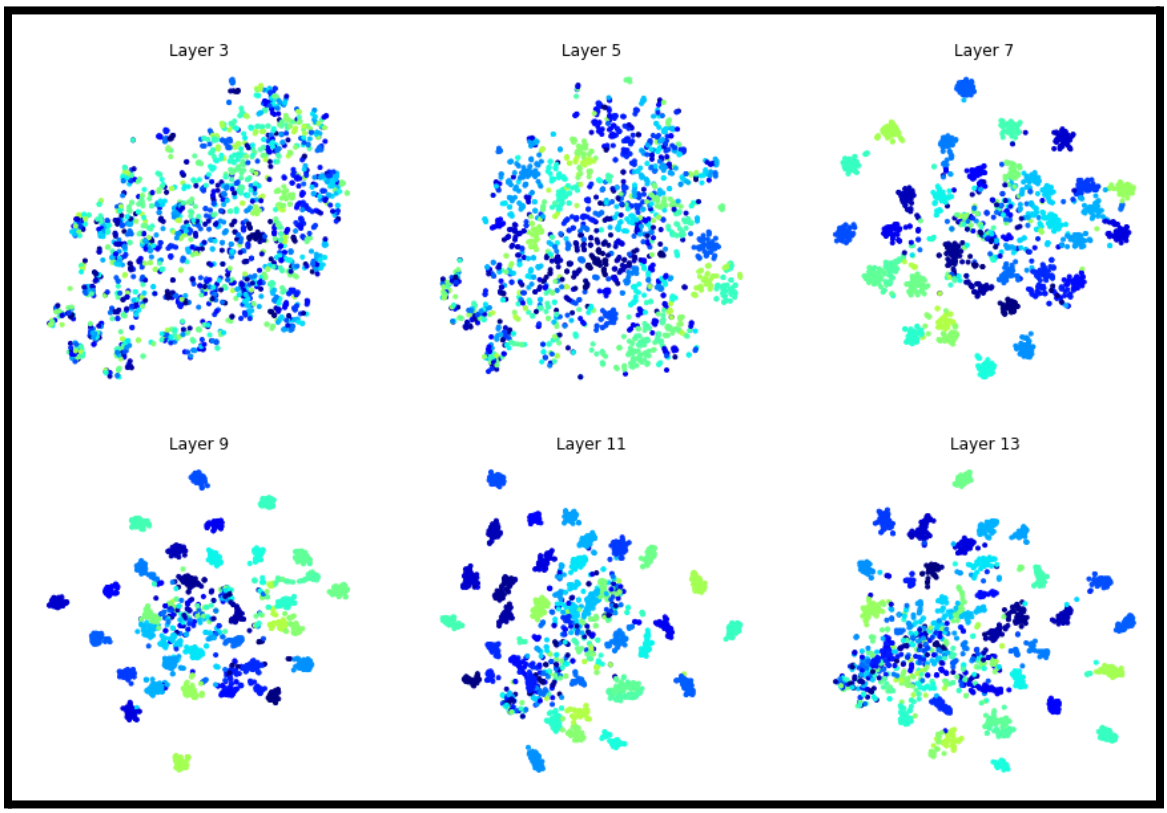


Figure 13: t-SNE command clusters through various hidden representations

Conclusion and future work

In this study, we have verified the fact by providing various evidence and examples that complex structures such as words are learned in deeper layers. We also qualitatively found out which layers learn what components(as in accent, phones, words) of an ASR system

We also gave a **JS divergence-based formulation** to quantify the similarities between the types of input(as in similar accents, etc). An important future direction can be verifying whether the JS divergence formulation mentioned works across all types of models and is able to provide reasonable explanations. Another direction for future work can be exploring whether the

layers also learn complex speech phenomena such as intent(which often involves looking back through time) and which part of the network captures these. These can also be done by post-hoc model agonistic explainability techniques such as LIME and perturbation-based techniques.

References

1. Chi, Po-Han, et al. "Audio albert: A lite bert for self-supervised learning of audio representation." *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021.
2. Belinkov, Yonatan, and James Glass. "Analyzing hidden representations in end-to-end automatic speech recognition systems." arXiv preprint arXiv:1709.04482 (2017).
3. Prasad, Archiki, and Preethi Jyothi. "How Accents Confound: Probing for Accent Information in End-to-End Speech Recognition Systems." Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020.
4. Shah, Jui, et al. "What all do audio transformer models hear? Probing Acoustic Representations for Language Delivery and its Structure." *arXiv preprint arXiv:2101.00387* (2021).
5. Correia, Gonçalo M., Vlad Niculae, and André FT Martins. "Adaptively sparse transformers." *arXiv preprint arXiv:1909.00015* (2019).
6. Lin, Jianhua. "Divergence measures based on the Shannon entropy." *IEEE Transactions on Information theory* 37.1 (1991): 145-151.
7. Haunschmid, Verena, Ethan Manilow, and Gerhard Widmer. "audiolime: Listenable explanations using source separation." *arXiv preprint arXiv:2008.00582* (2020).
8. Kominek, John, and Alan W. Black. "The CMU Arctic speech databases." *Fifth ISCA workshop on speech synthesis*. 2004.
9. Zue, Victor, Stephanie Seneff, and James Glass. "Speech database development at MIT: TIMIT and beyond." *Speech communication* 9.4 (1990): 351-356.
10. Warden, Pete. "Speech commands: A dataset for limited-vocabulary speech recognition." *arXiv preprint arXiv:1804.03209* (2018).
11. Van der Maaten, Laurens, and Geoffrey Hinton. "Visualizing data using t-SNE." *Journal of machine learning research* 9.11 (2008).