Audio Deepfake Detection

Shreeyut Maheshwari

April 1, 2025

1 Dataset

Dataset chosen was from huggingface *ArissBandoss/fake_or_real_dataset_for_rerecorded* which was only of 1.66Gb due to computational restriction.

2 Wave2vec 2.0 for classification(Implemented)

2.1 Key Technical Innovation

Wav2vec 2.0 is self supervised learning model that operates directly on raw audio waveforms rather than requiring labeled data or spectrograms.The architecture has a unique feauturized front end compromising of a stack of 1D CNN layers which operates on 16kHz audio waveforms.

2.2 Reoposted Performance Metrics

wav2vec2-xls-r-2b achieved an impressive 0.96% Equal Error Rate (EER) on deepfake detection tasks.However, the base wav2vec2 model achieved around 43.42% accuracy on OpenAI-generated deepfakes, showing some limitations with advanced TTS systems.

2.3 Why Approach is promising

The wav2vec approach is particularly promising because it can learn effective representations from unlabeled audio data, making it ideal for scenarios where labeled deepfake examples are limited.It also removes the need for feature engineering.

2.4 Challenges

- Low accuracy on TTS systems like from OpenAI
- The original model was designed for speech recognition rather than deepfake detection

3 Generalization Of Audio Deepfake Detection:ResNet

3.1 Key Technical Innovations

- Large Margin Cosine Loss:Reformulates softmax loss as cosine loss with a margin in cosine space maximize inter-class variance and minimize intra-class variance.
- Frequency Masking:A data Augmentation technique that randomly masks adjacent frequency channels during training by dropping out a consecutive frequency band range.
- Mean and Standard deviation Pooling
- Modified Filter Size and Strides:Preserves time resolution through convolutional layers and residual blocks, allowing the model to better capture temporal dynamics in audio signals.

3.2 Reported Performance Metrics

- Baseline ResNet18: 4.04% EER and 0.109 t-DCF on ASVspoof 2019 evaluation dataset.
- With LMCL: Reduced EER to 3.49% and t-DCF to 0.092.
- With LMCL + FreqAugmentation:Further EER to 1.81%

3.3 Why Approach is promising

- Generalization Ability
- Robustness
- Less Computation required than Wav2Vec

3.4 Potential Limitations and Challenges

- LMCL requires careful tuning of parameters
- EER increases from 0.017% to 6.186 %

4 Siamese Convolutional Neural Network Using Gaussian Probability Feature for Spoofing Speech Detection

4.1 Key Technical Innovations

- Uses two identical CNN branches processing features from genuine and spoofed GMMs
- Concatenates outputs of both branches for final classification

4.2 Reported Performances Metrics

- 0.157 EER and 0.00287 min-tDCF with CQCC feaures
- 0.710 EER and 0.19 min-tDCF with LFCC feaures

4.3 Why approach is promising

- Leverages complementary information from genuine and spoofed speech models
- Shows significant performances gains over single CNN approaches

4.4 Potential Limitations and Challenges

- Adversarial attacks

5 Documentation

5.1 Implementation

5.1.1 Challenges Encountered

- Computational Constraints to store training data and GPU for training the model
- Trying anything about audio for the first time and not familiar with the concepts for audio related deep learning like sampling rate,feature extractor.

5.1.2 Adressing the challenges

- For Computational restrictions smaller dataset was used instead from hugging face instead of standard datasets like ASV spoof and for GPU free colab GPU was used but because of less runtime available and deletion of data after runtime all the models were trained with Hugging Face and saved at Hugging Face.
- Learnt concepts like feature extractor and other audio-related deep learning concepts from online documentations.

5.1.3 Assumptions Made

As there was less time to implement thing the things which were mentioned in the papers were taken in consideration like Wav2Vec works well with less data.And as I had no knowledge of audio related stuff I had to trust very much on codes available on github repositories.

5.2 Analysis

5.2.1 Why you selected this particular model for implementation

Because of the computational constraints I couls not use a larger dataset so the paper of the wav2vec mentioned that the model performed very well on smaller datasets.

5.2.2 How the model works

Model is composed of a multi-layer convolutional feature encoder which takes the input as raw audio and outputs latent speech representations for T-time steps.Then they are fed into transformers to build representations capturing information from entire sequence.The output of the feature encoder is discreteized with a quantization to represent the targets in self-supervised objective and the self attention captures dependencies over the entire sequence of latent representation end-to-end.

Working of Individual Components

- **Feature Encoder** The encoder consists of several blocks containing a temporal convolutional followed by layer normalization and a GeLU activation function.The raw waveform input to the encoder is normalized to zero mean and unit variance.The total stride of the encoder determines the number of time-steps T which are input to the transformer.
- **Contextualized representation with Transformers** The output of the feature encoder is fed to a context network which follows the Transformer architecture.Instead of fixed positional embeddings which encode absolute positional information, a convolutional layer is used which acts as a positional embedding.Then output of the Convolutional followed by GELU to the inputs and then apply layer normalization.
- **Quantization Module** For self-supervised training we discretize the output of the feature encoder z to a finite set of speech representations via product quantization. Product quantization amounts to choosing quantized representations from multiple codebooks and concatenating them.

Original Training of the model

- **Masking** mask a portion of the feature encoder outputs.
- **Objective**

  $$\mathscr{L} = \mathscr{L}_m + \alpha \mathscr{L}_d$$

  The first one is Contrastive loss and other one is diversity loss and alpha is a tuned parameter.

5.2.3 Performance results on your chosen dataset

Here are the Wandb training logs

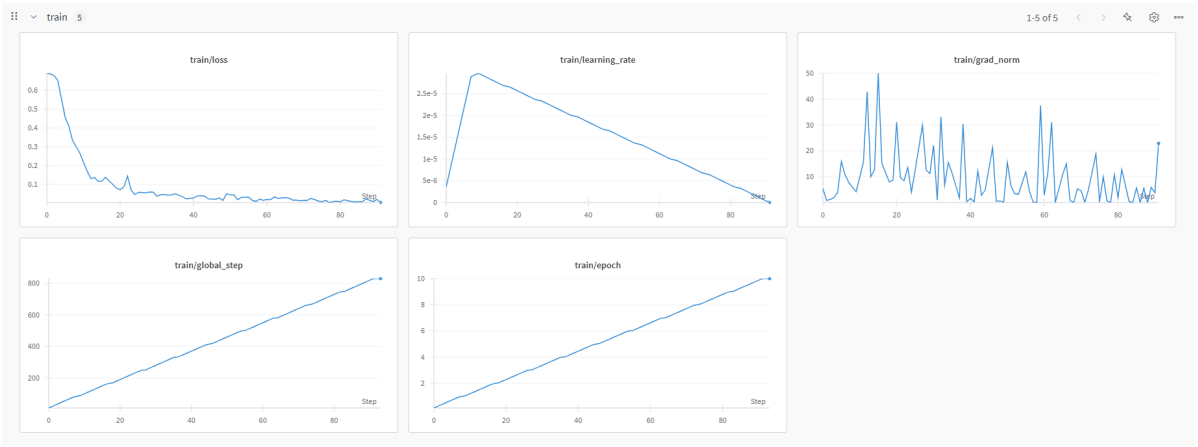

Figure 1: Training

Figure 2: Evaluation Dataset

5.2.4 Observed strengths and weaknesses

Though the model is performing good but it seems that the model is overfitting which is quite obvious because of the large model size and small dataset.Also model is not trained to handle adversarial attacks which is a huge negative point.

5.3 Suggestions for future improvements

Increasing the dataset and also adding different types of adversarial attacks which could maintain a overall generality of the model instead of overfitting and also preventing the model to rely on a specific frequency band.

5.4 Reflection questions

5.4.1 What were the most significant challenges in implementing this model?

The most significant challenge was to meet was training the model without having any prior experience to the audio deep learning feild and then also meeting the computational requirements.

5.5 How might this approach perform in real-world conditions vs. research datasets?

This approach have chances to perform lower in quality because the research dataset do not continously evolve with the elvoling TTS model which seem to be more closer to the real world speeches.

5.6 What additional data or resources would improve performance?

Certainly data from new evolving generative TTS model and along with adversarial attacks make the model robust but also it can not gurantee the loss convergence of the model because the model size could be a constraint in learning large amount of patterns.

5.7 How would you approach deploying this model in a production environment?

Hosting the model on a cloud service and with usage of REST API recieving the data and sending the prediction couldc be done