Comprehensive Implementation and Analysis: Siamese Convolutional Neural Network for Audio Deepfake Detection

Kartik Dua

05/04/2025

1. Implementation Process

Challenges Encountered

- **Dataset Compatibility:** The DECRO dataset provided audio files with varying durations and sampling rates. These inconsistencies required preprocessing and standardization before feeding them into the model.
- **Model Convergence:** The Siamese CNN architecture initially displayed slow convergence, and achieving a stable validation accuracy required multiple iterations.
- **Threshold Tuning:** Determining the threshold to classify similarity scores into binary labels (real vs. fake) proved non-trivial and involved extensive trial-and-error.
- **Time Constraints:** Implementing the model, performing optimization, and debugging the code within a tight deadline was challenging.

Solutions and Assumptions

- Preprocessing was standardized using Librosa to extract log-mel spectrograms from audio clips, resampled to a common sampling rate.
- Regularization techniques such as dropout and batch normalization were integrated to stabilize training and reduce overfitting.
- A threshold sweep between 0.1 and 0.9 was performed to empirically determine the optimal decision boundary. Although a threshold of 1.0 was initially used, the best dev accuracy was observed at a threshold of 0.8, which was then adopted.
- Cosine similarity was assumed to effectively capture relational differences between audio embeddings.

2. Analysis

Model Selection Justification

The Siamese Convolutional Neural Network (SCNN) was chosen due to its ability to compare pairs of data and produce a similarity score. This is highly appropriate for audio deepfake detection, where the authenticity of a sample can be learned through comparison to known real or fake samples. Unlike binary classifiers, this model type is more robust in scenarios where fake types are varied and evolving.

Technical Summary of the Model

- Each input pair is processed through two identical convolutional neural network branches, ensuring weight sharing.
- The CNN branches comprise convolutional layers with ReLU activation, batch normalization, and max pooling.
- Outputs from both branches are converted into fixed-dimensional embeddings.
- A cosine similarity layer computes the relational distance between embeddings.
- The similarity score is then compared against a threshold to determine whether a sample is real or fake.

Performance Evaluation on DECRO Dataset

- **Training and Dev Accuracy:** Best dev accuracy of **63.78%** was achieved at a similarity threshold of 0.8.
- **Test Set Accuracy:** Final evaluation on the unseen test set yielded **62.36%** accuracy.
- Accuracy degraded slightly at very low or very high threshold values, validating the model's sensitivity to boundary conditions.

Strengths and Weaknesses

- **Strengths:**
  - Robust in detecting similarity patterns rather than relying on absolute features.
  - Suitable for tasks with evolving adversarial inputs, like deepfakes.
  - Scalable architecture that generalizes well to other biometric verification tasks.
- **Weaknesses:**
  - Performance is sensitive to how the threshold is chosen.
  - Interpretation of cosine similarity lacks transparency compared to decision trees or SVMs.
  - Needs extensive tuning and validation for domain adaptation.
  - Training took over 4–5 hours even on a high-end machine equipped with a powerful GPU, indicating high computational demand.

Suggestions for Future Improvements

- Experiment with alternative distance metrics like Euclidean or Mahalanobis for comparison.
- Introduce attention-based pooling to allow the model to weigh time-frequency regions differently.
- Replace cosine similarity with a trainable similarity function via contrastive or triplet loss.
- Fine-tune the model using transfer learning from ASVspoof or larger audio deepfake datasets.

3. Reflection

a. Most Significant Challenges

- Redesigning the SCNN for audio-based data instead of visual inputs, which required adapting preprocessing and input dimensions.
- Building a balanced dataset of anchor-positive-negative audio pairs without introducing bias.
- Ensuring consistency in temporal features across samples with varied duration.

b. Real-world v/s Research Dataset Performance

- Real-world environments contain background noise, microphone variation, and signal distortion—all absent in curated datasets like DECRO.
- In production, the model would face adversarial examples not present during training, potentially degrading accuracy.
- Latency constraints and memory requirements may hinder deployment on edge devices.

c. Additional Data or Resources Needed

- Access to diverse audio deepfake datasets spanning languages, speakers, and spoofing techniques.
- Use of high-performance GPUs to explore deeper architectures and longer training cycles.
- Availability of real-world, labeled fake samples would enhance domain generalization.

d. Production Deployment Strategy

- Quantize the model for edge deployment while preserving accuracy.
- Develop an audio preprocessing module to normalize incoming data in real-time.
- Set up a feedback loop to collect user-verification mismatches and use them to retrain the model.
- Integrate with an alert system that triggers manual verification when the confidence score is low.