Overview

- The pairwise likelihoods of string-fret (S/F) combinations are estimated using a large collection of symbolic tablature [1].
- A novel inhibition loss incorporating the estimated likelihoods is proposed for deep learning based models.
- The output layer of a baseline guitar tablature transcription model [2] is re-formulated and augmented with the inhibition loss.

Guitar Tablature Transcription

Generate a 6-hot vector \( y_{i,s} \) for each frame \( i \) in a piece of audio, where \( s \) corresponds to the given fret class \( f \in \{ -1, 0, 1, \ldots, F \} \) for each string \( s \in \{1, \ldots, 6\} \). We use \( c \in \{1, \ldots, C\} \) interchangeably to denote combinations of string and fret (S/F), where \( C = 6 \times (F + 2) \).

\[
\begin{array}{cccccccccccc}
6 & - & E & A & D & G & B & e & E & A & D & G & B & e
\end{array}
\]

Output Layer Formulation

Contemporary tablature transcription models [2, 3] apply the softmax activation across fret classes for each string at the output layer.

\[
L_{CCE} = - \frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} \log(y_{i,s,c,n}) + (1 - y_{i,s,c,n}) \log(1 - z_{i,s,n})
\]  

This treats transcription as 6 independent classification tasks, ignoring the typically high correlation between the S/F combinations making up a fingering. We re-formulate the output layer using sigmoid activations, allowing us to introduce a novel inhibition loss.

\[
L_{BCE} = - \frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} z_{i,s,c,n} \log(y_{i,s,c,n}) + (1 - z_{i,s,c,n}) \log(1 - y_{i,s,c,n})
\]

Datasets

- Large collection of GuitarPro files featuring tablature for many popular full-length songs.
- Includes artists spanning many musical styles, with a lean toward rock and metal.
- We process all guitar tracks in standard tuning, yielding 39967 pieces of symbolic tablature.

GuitarSet [4]

- Contains roughly 3 hours of acoustic guitar audio with string-level note annotations.
- Features 6 guitarists playing 2 unique interpretations over 30 different chord progressions.

Estimating Pairwise Likelihood

We can estimate the pairwise likelihood of two S/F combinations \( c_i \) and \( c_j \) using an arbitrary collection of symbolic tablature data (e.g., DadaGP [1]). Given the symbolic tablature for a single track, we compute the intersection over union (IoU) of frame-level occurrences for all pairs of S/F combinations.

\[
\text{inter}(i,j) = \sum_{n=1}^{N} t_i \land t_j, \quad \text{union}(i,j) = \sum_{n=1}^{N} t_i \lor t_j
\]

Let \( T'(i,j) \) be the set of tracks where \( c_i \) and \( c_j \) independently, each occur in at least one frame. The IoU of the pair is averaged across these valid tracks.

\[
\text{IoU}(i,j) = \frac{1}{|T'(i,j)|} \sum_{n \in (t_i \land t_j)} \text{inter}(i,j), \quad \text{union}(i,j)
\]

where \( |T'(i,j)| \) is the cardinality of \( T'(i,j) \). Note this is only valid for pairs where \( |T'(i,j)| > 0 \). We set \( \text{IoU}(i,j) = 0 \) for all pairs where \( |T'(i,j)| = 0 \).

Inhibition Loss

We introduce a novel loss term to inhibit the co-activation of unlikely pairs:

\[
L_{inh} = \frac{1}{2N} \sum_{n=1}^{N} \sum_{i=1}^{C} \sum_{j=1}^{C} z_{i,s,c,n} w(c_i, c_j).
\]

The product for every combination of activations is taken and scaled by an inhibition weight, a penalty between 0 and 1 for producing high activations for the combinations in the pair in a single frame. The result is summed over all combinations. We set the inhibition weights to be the complement of the pairwise likelihoods estimated using Equation (4), boosted with parameter \( b \).

\[
w(c_i, c_j) = (1 - \text{IoU}(i,j))^b
\]

Including a scaling term \( A \) for balancing the two terms, the total loss becomes

\[
L_{total} = L_{BCE} + A L_{inh}
\]

Experiments

- Train and evaluate on GuitarSet [4] following 6-fold cross-validation schema [2].
- Experiment with holding out an extra dataset split for validation.
- Experiment with inserting a uni-directional LSTM before the output layer.
- Experiment with variations of the proposed output layer formulation.
- Adopt the metrics proposed in [2], but average across tracks, then folds.
- Compute inhibition losses \( L_{inh} = b \) and \( L_{inh}^+ = b \) on final predictions.
- Count number of duplicate pitch \( E_{dp} \) and false alarm \( E_{fa} \) errors.

Acknowledgements & References

This work is partially funded by National Science Foundation grants IIS-1446104 and DGE-1922591. All of the code is available at https://github.com/air-audio/uitar-transcription-with-inhibition.