Interpreting music genre classification in CNNs

- Johnny Ren, David Moon, Clara Guo

Introduction

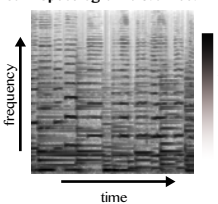
The importance of model interpretability has become increasingly obvious as DL is being applied to more and more human-centric issues and many believe in and ensure the right to an explanation. We investigate different techniques from a variety of papers to interpret filter activations as well as modifcations to architecture (e.g. masking and different pooling methods) intended to make these activations more intuitively meaningful. These papers are by Zhang et al., Choi et al., and Kim et al.

Methodology

Preprocessing:

We converted the GTZAN dataset—comprised of 100 30 second wav files per genre (10 genres total)—into log scaled mel spectrograms using the librosa library. Each wav file was turned into 10 spectrograms of roughly 3 seconds each. These spectrograms were what was passed into the CNN. We used an 80:10:10 split for the train data, validate data, and test data ratio, respectively.

Jazz Spectrogram Visualized:



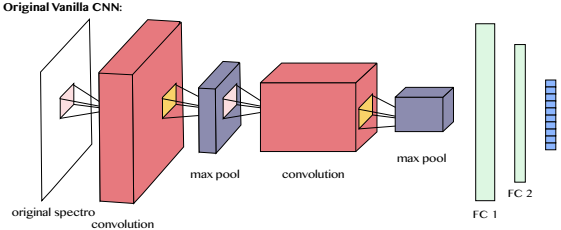
Architecture:

We created multiple models to compare the effects of different interpretability techniques. The original model is a vanilla CNN with 2 convolution layers followed by 2 dense. The masked model applies an elliptical masking function to the output of the 2nd convolutional layer. The third model implements global average pooling following 3 conv layers and 1 conv transpose to produce activation maps.

Visualization and Auralisation:

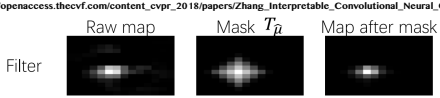
Spectogram values were scaled to the range [0,255] to create BW png representations. Deconvolution was computed by upsampling and performing a convolution with flipped filters that were transposed to reverse the in and out channel dimensions. Conv 1 activations were calculated using the output using the model's weights and then creating a mask from the filter activation to elementwise multiply by the input. To convert to wav files again, a librosa built-in was used. Class activation maps were computed using a weighted sum of filter activations obtanied from global average pooling and visualized using the OpenCV library and matplotlib.

Original Vanilla CNN:



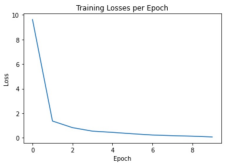
Visualization of Masking (taken from Zhang et al.):

https://openaccess.thecvf.com/content_cvpr_2018/papers/Zhang_Interpretable_Convolutional_Neural_CVPR_2018_paper.pdf



Results

Vanilla:

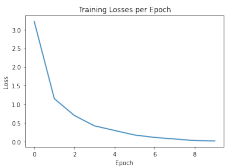


Validate accuracy: 67.08%

Test accuracy: 68.23%

Training accuracy across 10 epochs: 77.38%

Masked:

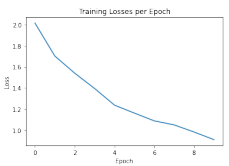


Validate accuracy: 68.44%

Test accuracy: 69.90%

Training accuracy across 10 epochs: 83.01%

Global Avg Pooling:



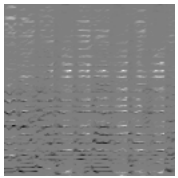
Validate accuracy: 68.23%

Test accuracy: 68.12%

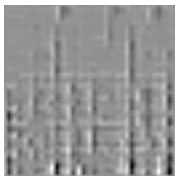
Training accuracy across 10 epochs: 51.55%

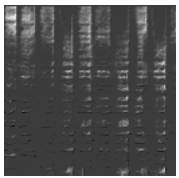
Spectrograms for Hit Me Baby One More Time by Britney Spears:

Conv 1 Activations (Filter 16):

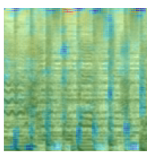


Deconvolved Activations (Layer 2):

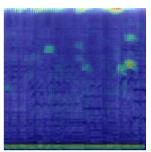




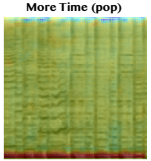
Class Activation Maps:



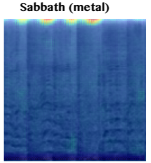
Hit Me Baby One More Time (pop)



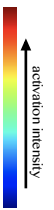
Paranoid by Black Sabbath (metal)



Is This Love by Bob Marley (reggae)



Another One Bites The Dust by Queen (rock)



Discussion

Challenges/Limitations:

- Accuracy was extremely sensitive to filter size
- The original paper we planned to implement did not provide enough specifics about their loss implementation for us to recreate it. Because of this, we pivoted to exploring global average pooling instead
- Global average pooling requires no other dense layers to preserve the ability to create activation maps, restricting the functions the GAP model could learn and reducing our accuracy
- Convolution isn't invertible, so upsampling was used to increase dimension size for approximate deconvolution

Future Work:

- Implementing a custom loss that ensures each high level filter learns features unique to each class
- Experimenting with different filter and mask shapes to explore the significance of a larger breadth of time vs frequencies
- Adding more convolutional layers to explore the different types of features filters pick up in low vs higher layers