Music Latent Representation and Disentanglement for Genre Classification

CSCI 699 (Representation Learning: Theory and Practice)
Beyond-Assignment Project

Ehsan Qasemi qasemi@usc.edu
Binh Vu binhlvu@usc.edu
Minh Pham minhpham@usc.edu
Basel Shbita shbita@usc.edu

December 15, 2019

Abstract

Music is an expressive form of art that is generally perceived as the most universal of all art forms. It combines vocal or instrumental sounds in a harmonious and expressive way. A significant amount of information lies in a musical composition through variations of different properties (e.g., tempo). In this project we investigate the use of unsupervised representation learning techniques to compress musical samples into a low-dimensional representation and use it for the task of music genre classification. In our study, we aim to facilitate the robust learning of disentangled representations (i.e., features like genre) by increasing the information capacity of the latent code during training. We perform the training and evaluation of our models using the FMA public dataset[1].

1 Motivation

Music is the art of combining tones to form an expressive composition; one that involves combinations of pitch, timbre, rhythm, dynamics, tempo, texture, melody and harmony. Putting these elements together in various ways creates a huge diversity of music – from African drumming to classical music. Some of these features control acoustic structure of the sound wave, especially the regularity or rate of repetition. For example, duration refers to the length of the tone, while dynamics refers to how loud or quiet a note is.

Music genres are categories that have arisen through a complex interplay of cultures, artists, and market forces to characterize similarities between compositions and organize music collections. Generally, no universal genre taxonomy exists; the boundaries between genres remain blurred, making the problem of music genre classification a nontrivial task. We consider a genre as a category consisting of songs sharing certain aspects of musical characteristics.

Deep neural networks, and in particular those trained in an unsupervised way such as AEs (autoencoders) or GANs (generative adversarial networks), have shown nice properties to extract latent representations from large and complex datasets. β-VAEs (β-variational autoencoders) can be seen as a probabilistic autoencoders that deliver a parametric model of the data distribution. These models encourage the latent coefficients to be mutually orthogonal and lie on a similar range. Such properties may be of potential interest for using the extracted latent coefficients as control parameters for a music generation or classification tasks.

By projecting the signal data from the signal space into a low-dimensional latent space (encoding or embedding) using a β-VAE model, we believe it is possible to achieve a high correlation between the extracted dimension coefficients and the observed properties of the data (i.e., genre). In our work, we investigate several approaches to encode unsupervised representations learnt on music samples to classify its genre.

[1] https://github.com/mdeff/fma

2 Related Work

Engel et al. [1] introduced NSynth, an audio synthesis method that is based on a time-domain autoencoder inspired from the WaveNet speech synthesizer [2]. The authors investigated the use of this model to find a high-level latent space well-suited for interpolation between instruments. Their autoencoder is conditioned on pitch and is fed with raw audio from their large-scale multi-instrument and multi-pitch database (the NSynth dataset). This approach led to promising results but has a high computational cost.

Roche et al. [3] presented a study investigating the use of non-linear unsupervised dimensionality reduction techniques (i.e., AEs, DAEs, LSTM-AEs, VAEs and PCA) to compress a music dataset into a low-dimensional representation which can be used in turn for the synthesis of new sounds. Their experiments were also conducted on the NSynth dataset.

Both of these studies are based on short samples from individual instruments while we are focusing on samples drawn from fully-composed and instrument-rich song samples. Our goals differ since we are trying to capture information that is more beneficial for the classification of genre (note structure, rhythm and melodies rather than tone color or notes that can be classified as single-instrument features). Nonetheless, this line of work has shown interesting preliminary results; as well as we were inspired by some of the techniques and suggestions found in their work which we have utilized and found useful in our project.

Kim et al. [4] utilized a transfer learning framework, learning artist-related information that was used at inference time for genre classification. This work differs from ours since they explore additional features for the purpose of genre classification only, without considering a dense representation of the data that can be used for reconstruction. We aim to use a latent representation that is "reconstructable" and can be used for the task of genre classification.

3 Dataset

A labeled dataset was used for training and testing our models. The dataset was originally constructed from the Free Music Archive (FMA), an interactive library of high-quality, legal audio downloads directed by WFMU[2]. The FMA provides as much as 343 days of Creative Commons-licensed audio from 106,574 tracks from 16,341 artists and 14,854 albums, arranged in a hierarchical taxonomy of 161 genres. It provides full-length and high-quality audio, pre-computed features, together with track- and user-level metadata and tags. For the purpose of our project we used a subset of the data (`fma_small.zip`) that includes 8,000 tracks of 30s covering 8 balanced genres.

4 Data Pre-processing

Prior to setting our models and experiments, we performed several steps to pre-process the data and reduce the complexity of our task.

In sound processing, the Mel Spectrogram make up a representation of the short-term power spectrum of a sound. Mathematically speaking, it is the result of some non-linear transformation of the frequency scale. In order to generate the Mel Spectrogram we first separate the audio signals to windows and compute the FFT (Fast Fourier Transform) for each window to transform from the time domain to the frequency domain. Next, we generate the Mel-scale by taking the entire frequency spectrum, and separating it into evenly spaced frequencies. We then decompose the magnitude of the signal into its components, corresponding to the frequencies in the Mel-scale for each window. Using the amplitudes of the resulting spectrum allows us to create a representation of the data that is more close to how the human's auditory system operates.

Additionally, for computation-cost reasons, we split the music files into smaller file chunks with equal length. This causes the recurrent layers to have shorter sequences to process. Due to fixed length of the inputs, the loss function is significantly simpler (handling variable length sequences in a batch requires additional masking in the loss function).

[2] `https://wfmu.org/`

5 Architecture

As part of our study, we investigate several state-of-the-art architectures applied on music samples, including naive Seq2Seq in subsection 5.1, step-based β-VAE in subsection 5.2, and Wavenet in subsection 5.4. In addition to using raw music files, we have an additional Seq2Seq model using Mel Spectrogram of the data in the music sample file as input in subsection 5.3.

5.1 Naive Seq2Seq

As our first model, we used the naive sequence-to-sequence architecture [5]. Seq2Seq architecture is about training models to convert a sequence from one domain (e.g. a source language) to a sequence in another domain (e.g. a target language). In our work, we used the Seq2Seq as an autoencoder where the source and target sequences are the same and the model's task is to learn a latent representation in its bottleneck for the music sample, which is here processed as a sequence in time domain.

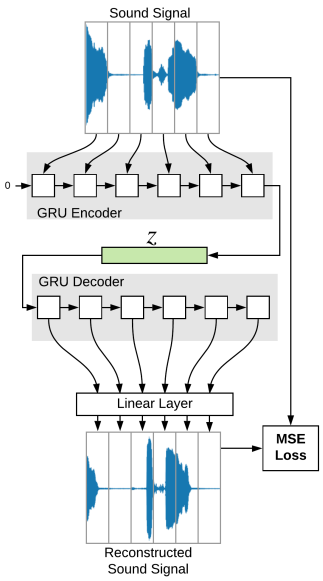


Figure 1: Naive Seq2Seq architecture

Our Seq2Seq model (depicted in Figure 1) consists of two GRU (Gated Recurrent Unit) layers. The first GRU works as the encoder to go from the raw music sample to the latent bottleneck representation in the middle. The decoder consists of the second GRU layer and a linear layer that starts from the latent representation and generates the input music sample (one sample at a time). Additionally, to improve the convergence, we trained our model using the "teacher forcing" method.

We trained our model with three hyperparameter setups: 1) $GRU_{dim} = Z_{dim} = 8$, 2) $GRU_{dim} = Z_{dim} = 16$, and 3) $GRU_{dim} = Z_{dim} = 32$. However, we did not notice a significant improvement in any over the rest.

5.2 Step-based β-VAE

Our step-based β-VAE is inspired by the original β-VAE model that was used for images in HW3. The Mel-scale Spectrogram after preprocessing has the shape of (B, L, C) where B is batch size, L is the length of the signal and C is the number of components. In this model, we consider each time step in the spectrogram as one reconstructing sample with the shape of $(C, 1)$. In the encoder module, we use multiple `Conv1D` layers to compress the input vector to our latent vector z and then use multiple `Conv1DTranspose` layers to reconstruct the original signal. Finally, all of the reconstructed timesteps will be concatenated into the final spectrogram. Figure 2 shows the overview of our step-based β-VAE model. The dimension of the latent variable was 100.

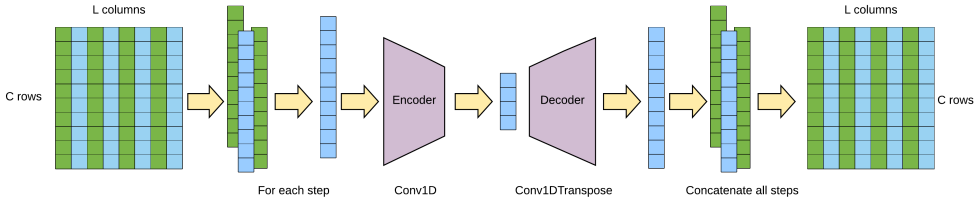


Figure 2: Step-based VAE model

5.3 Seq2Seq Autoencoder on Mel Spectrogram

In this sequence to sequence autoencoder (Seq2Seq-AE), we use an LSTM model to transform the Mel Spectrogram of the samples to a vector of size 512, we then use another LSTM model to reconstruct the original Mel Spectrogram from the vector (as depicted in Figure 3). The objective function is the reconstruction loss (Mean Square Error) between the original and the reconstructed Mel Spectrogram.

5.4 Wavenet

Raw audio samples contain an enormous amount of information about the songs. We have to use approximately 500,000 data points to represent 30 seconds. Thus, it would be hard for a model to capture the long dependency between data points in a very long sequence without lots of data labels. In order to address this, we use a Wavenet model [2] that is trained on raw audio points to predict the next point: $P(x_t|x_1, ..., x_{t-1})$. Figure 4 shows the overview of the Wavenet architecture that we use, which is similar to the original model. In addition to the raw audio samples, we also condition the input distribution on the Mel Spectrogram to help the prediction. The Mel Spectrogram is transformed through a list of upsampling layers to make it match the length of the audio sequence.

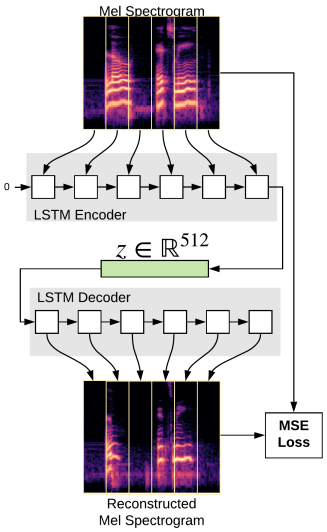


Figure 3: LSTM autoencoder architecture

After training, the outputs of the layer before the prediction layer (the last layer) contain information that assists predicting $P(x_t|x_1, ..., x_{t-1})$. Hence, they can be used as embedding vectors or latent representations of data points. The embedding vectors should capture the long-range dependency of the current data point with its previous points, which is the desired property. To use the embedding vectors in classifying music genres, a common choice in the NLP community is to average those vectors. Training a Wavenet model is hard and time-consuming due to the enormous amount of dataset. It takes more than 7 days to train just 50 epochs on the FMA small dataset. Therefore, we have not been able to report the result of the Wavenet model as to the time of the publication of this report.

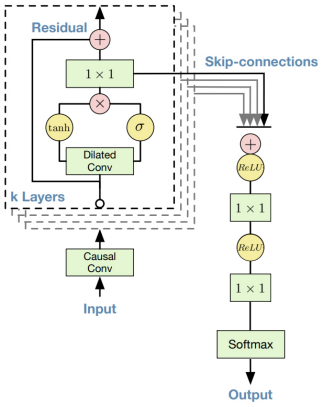


Figure 4: Wavenet architecture

6 Experiments

6.1 Supervised Genre Prediction

In this experiment, we first train each model to learn and extract latent representations of a song. Then, we train a classifier (Random Forest classifier of 200 estimators and max depth of 20 or a Logistic Regression classifier) to predict genres from the latent representations.

We report the accuracy of different models in Table 1. Note that we also trained a supervised CNN model directly on the Mel Spectrogram with labeled data as a baseline (first row). Even though the accuracies of the step-based β-VAE and the Seq2Seq-AE are around 0.3, their performance is only 0.1 less than the supervised model and higher than a random guess (0.125).

Table 1: Accuracy of different models

Model	Train on	Train accuracy	Test accuracy
Supervised CNN	data	0.58	0.43
Step-based β-VAE	latent	0.95	0.30
Seq2Seq-AE-Mel	latent	0.99	0.32
Naive Seq2Seq-AE	latent	0.85	0.45

6.2 Reconstruction

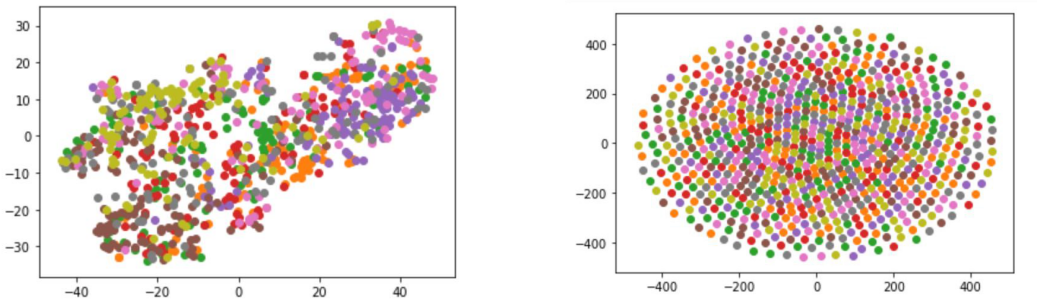
We also report our reconstruction error to evaluate the performance of our representation learning models (Table 2). The two models have reasonable reconstruction error. However, the Seq2Seq-AE model has a much higher error than the step-based β-VAE model. One possible reason is that the LSTM model does not work well with long sequences, which is well known in the machine learning community.

Table 2: Reconstruction error of different models

Model	Reconstruction Error
Seq2Seq-AE	0.66
Step-based β-VAE	0.03

6.3 Latent Space Visualization

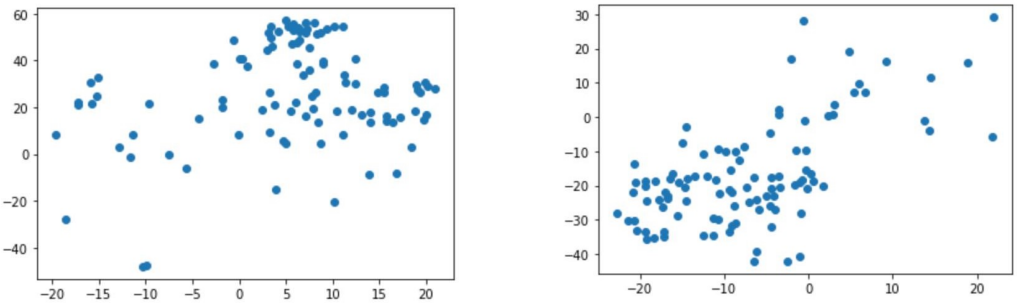
We show the t-SNE charts of the latent spaces with regard to the genre labels of two models: Seq2Seq-AE and β-VAE.



(a) Seq2Seq-AE latent space

(b) β-VAE latent space

Figure 5: Latent space of representation learning models



(a) Instrumental genre

(b) Hip-hop genre

Figure 6: Latent space of two genres: Hip-hop and Instrumental

From Figure 5a, we can see that there are two genres that are quite clustered together in the Seq2Seq-AE latent space: hip-hop (brown) and instrumental (purple). Details of these two genre are shown in Figure 6. On the other hand, it is clearly that our β-VAE model still need to be tuned until it is ready for the genre classification task.

7 Discussion and Conclusion

Training an autoencoder so that it captures a meaningful representation of the input is difficult. Our experiments have shown that capturing a latent representation of a sequence-based structures like songs are much more difficult compared to images. Moreover, it is hard to disentangle songs' latent space into components such as genres, pitches, rhythm since they are correlated with others. In our proposed models, there are lots of model hyper-parameters to fine tune in NN based models, including learning rate, weight decay, momentum, and other forms of regularization which may be subject of future studies, however our results show that their effect are not significant. It must be noted that we cannot disregard the fact that the hyper-parameters chosen for the models were probably inappropriate. Further work is needed to examine how these models might be improved.

References

[1] Engel, Jesse, et al. "Neural audio synthesis of musical notes with wavenet autoencoders." *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017.

[2] Oord, Aaron van den, et al. "Wavenet: A generative model for raw audio." *arXiv preprint arXiv:1609.03499* (2016).

[3] Roche, Fanny, et al. "Autoencoders for music sound modeling: a comparison of linear, shallow, deep, recurrent and variational models." *arXiv preprint arXiv:1806.04096* (2018).

[4] Kim, Jaehun, et al. "Transfer learning of artist group factors to musical genre classification." *Companion Proceedings of the The Web Conference 2018*. International World Wide Web Conferences Steering Committee, 2018.

[5] Sutskever, I., O. Vinyals, and Q. V. Le. "Sequence to sequence learning with neural networks." *Advances in NIPS* (2014).